

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra měřicí a řídicí techniky

## BAKALÁŘSKÁ PRÁCE

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra měřicí a řídicí techniky

**Systém pro automatickou detekci převrnutí  
invalidního vozíku  
Automatic Wheelchair overturn Detection System**

## Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

V Ostravě, dne .....  
podpis

## **Poděkování**

*Děkuji především vedoucímu bakalářské práce panu Ing. Martinu Černému za vedení bakalářské práce a za cenné rady při její realizaci. Dále děkuji svým rodičům a příbuzným za podporu a trpělivost během tvorby této bakalářské práce.*



## Abstrakt

Bakalářská práce se zabývá návrhem hardwaru a softwaru pro vyhodnocení stavu invalidního vozíku. Stavem invalidního vozíku se myslí jeho běžná pracovní poloha (na kolečkách), nebo stav při převrnutí (na boku...) s tím, že alarmující stav je při převrnutí.

Jádrem zařízení pro měření náklonu vozíku je tříosý akcelerometr MMA7660FC od firmy Freescale. Je to akcelerometr miniaturních rozměrů založený na technologii MEMS. Pro zpracování naměřených údajů a odeslání dat do počítače se využil mikroprocesor PAN4551 od firmy Panasonic. Pro bezdrátovou komunikaci akcelerometru s počítačem se využila technologie ZigBee.

Tato práce by měla pomoci v oblasti zdravotní prevence a osobní bezpečnosti. Měla by pomoci invalidním osobám kteří z důvodu ochrnutí, nemoci...aj. nejsou schopni chůze a jsou tak závislí na invalidním vozíku. S tím jsou spojeny problémy jako je monitorování invalidního vozíku v exteriéru (jestli se vozíčkář nepřevrhl, jestli neupadl...)

## Klíčová slova

akcelerometr, invalidní vozík, převrnutí, zigbee, mikroprocesor

## Abstract

The bachelor thesis deals with hardware and software for evaluating the status of the wheelchair. State of the wheelchair can be Its normal operating position (on wheels), or the state of overturn (on the side ...), with that the alarming situation is when Its overturned.

The main device for measuring the tilt of the wheelchair is 3-axial accelerometer MMA7660FC from Freescale. It is a miniature accelerometer based on the MEMS technology. For processing the measured data and sending data to a computer microprocessor PAN4551 is used, from Panasonic too. I use ZigBee technologies for wireless communication between accelerometer and computer.

This work should help in the health prevention and personal safety. It should help persons who, because of paralysis, illness, etc. ... are unable to walk and are dependent on a wheelchair. With the associated problems such as monitoring a wheelchair outside (if the person on the wheelchair did not overturn, fall ...)

## Keywords

accelerometer, wheelchair, overturn, zigbee, microprocessor

## Seznam použitých symbolů a zkratek

GPS	Global Positioning Systém - polohový systém
ACK	Acknowledge - potvrzení
ADC	Analog to digital converter - analog digital převodník
AMP	amplifier - zesilovač
APL	A Programming Language - programovací jazyk
AVDD	analog supply voltage - analogové napájení
BDM	Background Debugging Module - programátor BDM
BiCMOS	bipolar and metal–oxide–semiconductor - prvky bipolární technologie i technologie CMOS
CFP	Contention Free Period - interval kdy nedochází k soupeření o přístup k médiu
CMOS	Complementary Metal–Oxide–Semiconductor - výrobní postup procesoru
CPU	Central Proccesor Unit - procesor
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance - pravděpodobnostní protokol přístupu k médiu
DVDD	digital supply voltage - napájení digitální části
FFD	Full Functional Device - zařízení implementují kompletní protokolový rámec
GND	Ground - zem
GPIO	General Purpose Input/Output - piny, které mohou být konfigurovány jako vstupy i jako výstupy
GTS	Guaranteed Time Slot - vyhrazení pro pomalá a prioritní zařízení
I2C	Inter-IC-bus - komunikační standard
IEEE	Institute of Electrical and Electronics Engineers - institut pro elektrotechnické a elektronické inženýrství
INT	Interrupt - přerušení
LED	Light-emitting diode - Světlo vyzařující dioda
MEMS	Micro Elektro Mechanical Systém - mikro mechanické systémy
NWK	Network - síť
PAN ID	Personal Area Network identifier - adresa v síti
R	read - čtení
RAM	random access memory - přístupová paměť
RFD	reduced-functionality device - způsob implementace protokolové knihovny
RS232	Recommended Standard 232 - komunikační standard
SCL	Serial Clock - časová linka
SDA	Seriál Data - datová linka
Si	Silicium - křemík
SP	Stop – ukončení
ST	Start – začátek
TTL	transistor-transistor-logic - způsob provedení logiky integrovaných obvodů
USB	Universal Serial Bus - komunikační standard
ZDO	ZigBee Device Object - Zigbee objekty

# Obsah

<b>Úvod.....</b>	<b>- 2 -</b>
<b>1 MĚŘENÍ NÁKLONU.....</b>	<b>- 3 -</b>
1.1 GYROSKOPY.....	- 3 -
1.2 INKLINOMETRY.....	- 3 -
1.3 RTUŤOVÉ SNÍMAČE.....	- 4 -
1.4 AKCELEROMETRY.....	- 4 -
1.4.1 Akcelerometry se seismickou hmotou.....	- 5 -
1.4.2 Tepelné akcelerometry.....	- 7 -
1.4.3 MEMS kapacitní akcelerometry.....	- 8 -
<b>2 BEZDRÁTOVÁ KOMUNIKACE.....</b>	<b>- 10 -</b>
2.1 TECHNOLOGIE ZIGBEE.....	- 10 -
2.2 VZNIK ZIGBEE A JEHO VLASTNOSTI.....	- 10 -
2.3 SPECIFIKACE RADIOVÉ ČÁSTI STANDARDU.....	- 10 -
2.4 TOPOLOGIE SÍTĚ ZIGBEE.....	- 11 -
2.5 SPOTŘEBA ZAŘÍZENÍ.....	- 12 -
2.6 STRUKTURA PROTOKOLŮ ZIGBEE.....	- 12 -
<b>3 NÁVRH ZAŘÍZENÍ PRO DETEKCI PŘEVHRNUTÍ.....</b>	<b>- 14 -</b>
3.1 MMA7660FC.....	- 14 -
3.2 SC18IM700.....	- 18 -
3.3 PAN4551.....	- 19 -
3.3.1 Mikroprocesor MC9S08GT60.....	- 20 -
3.4 VÝVOJOVÁ DESKA EVALKIT PAN4551.....	- 21 -
<b>4 REALIZACE SYSTÉMU.....</b>	<b>- 22 -</b>
4.1 ZHOTOVENÍ TESTOVACÍHO MODULU.....	- 22 -
4.2 POPIS KOMUNIKAČNÍHO PROTOKOLU.....	- 23 -
4.3 OBSLUHA AKCELEROMETRU.....	- 25 -
4.4 PROGRAM PRO MCU.....	- 29 -
4.5 DOSAH ZAŘÍZENÍ.....	- 30 -
<b>5 VIZUALIZACE PŘEVHRNUTÍ.....</b>	<b>- 31 -</b>
<b>ZÁVĚR.....</b>	<b>- 32 -</b>
<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>- 33 -</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>- 34 -</b>

## Úvod

... Vozíčkář (fyzicky i lehce psychicky postižená osoba) na odpolední podzimní vyjížďce do okolí rehabilitačního ústavu. V nejvzdálenější části vyjížďkové trasy najíždí na terénní nerovnost, vozíček je převrhnut a vozíčkář vypadá z vozíku. Nemůže se dostat do vozíku, kde má mobilní telefon, nikdo neslyší jeho volání o pomoc. Je nalezen až pozdě večer ve špatném fyzickém stavu – prochlazen.

Cílem této práce je tedy systém pro automatickou detekci převrnutí vozíku. Princip řešení spočívá v tom, že pro vyhodnocení stavu invalidního vozíku se použil senzor zrychlení – akcelerometr MMA7660 od firmy Freescale. Senzor se nastavil do režimu, kdy automaticky ukládá informace o poloze do registru \$03. Informace jsou dále přijaty a zpracovány mikroprocesorem MC9S08GT60 který má komplexní řešení bezdrátové ZigBee komunikace. Po vyhodnocení převrnutí jsou data odeslána do přijímací stanice, ta po kontrole paketů odešle informace na sériovou linku do počítače.

Bakalářskou práci jsem rozdělil do pěti kapitol. V první kapitole jsem popsal různé způsoby měření náklonu a možnosti řešení práce s jinými senzory náklonu.

Druhá kapitola se věnuje bezdrátové komunikaci, přesněji technologii Zigbee, popisuje její technologii, vznik a princip fungování.

Ve třetí kapitole se zabývám návrhem řešení práce, popisují jednotlivé součástky a mimo jiné klíčovou součástku této práce a způsob, jak komunikuje s okolním hardwarem – akcelerometr MMA7660. Také jsem popsal součástky a moduly nutné pro zpracování naměřených dat.

Další kapitola podrobně popisuje komunikaci, testování akcelerometru a komunikačních modulů. Představuje princip programu pro vyhodnocení převrnutí vozíku. Také zde představuji finální desku plošných spojů, její součástky a funkci.

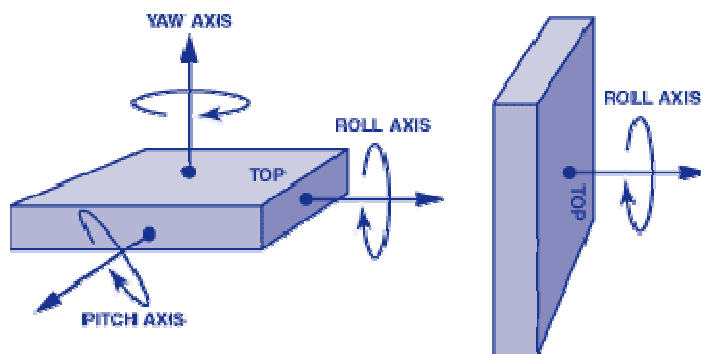
Poslední kapitola je popis vizualizačního software. V závěru jsou zhodnoceny výsledky bakalářské práce.

# 1 Měření náklonu

Pro monitoring polohy vozíku je třeba snímač náklonu. Pro snímání náklonu se běžně používají gyroskopy, inklinometry, rtuťové snímače nebo akcelerometry. Senzory se od sebe liší konstrukcí, principem měření, způsobem komunikace, velikostí, cenou...

## 1.1 Gyroskopy

Vývoj moderních gyroskopů se datuje od 70.let 20. století kdy vznikly potřeby zdokonalovat přesnost dopadu balistických raket. Dnes se gyroskopy nejčastěji využívají v navigaci u letadel a určování pozice přes GPS systémy. Princip měření spočívá v setrvačnicku který udržuje osu své rotace v inerciálním prostoru. Přesnost gyroskopu je závislá na schopnosti udržení stability jeho otáček. Mohou být také využity ke stabilizaci obrazu ve fotoaparátech, dalekohledech apod. přístrojích. Dříve se gyroskopy skládaly jen z mechanických částí, dnes však obsahují kromě mechanického měřicího systému i vyhodnocovací elektronické obvody, logické systémy. Výstupy mohou být analogové, digitální nebo obojí.

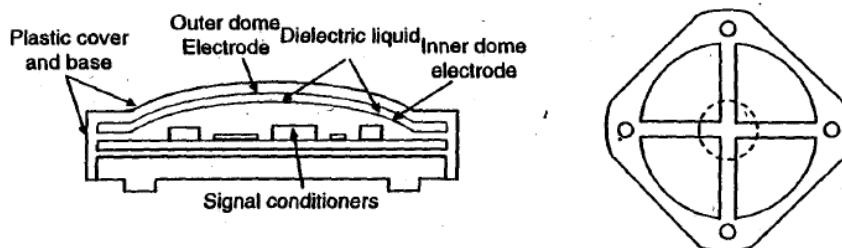


Obr. 1 Možnosti měření natočení a rotace

[3]

## 1.2 Inklinometry

Inklinometry (náklonoměry), jak již z názvu vypovídá, jsou senzory měření inklinace. Jsou to vlastně akcelerometry v módu měření inklinace. Princip činnosti inklinometru spočívá ve změně kapacity mezi protilehlými elektrodami při změně skonu kapaliny s velkou permitivitou v měřicí komůrce. Rozlišovací schopnost je  $0,01^\circ$ , rozsah úhlu je  $\pm 20^\circ$ . Nevýhodou je pomalé ustálení kapaliny, pomalé měření. Pro zobrazení náklonu na počítači se nejčastěji vizualizuje umělý horizont.

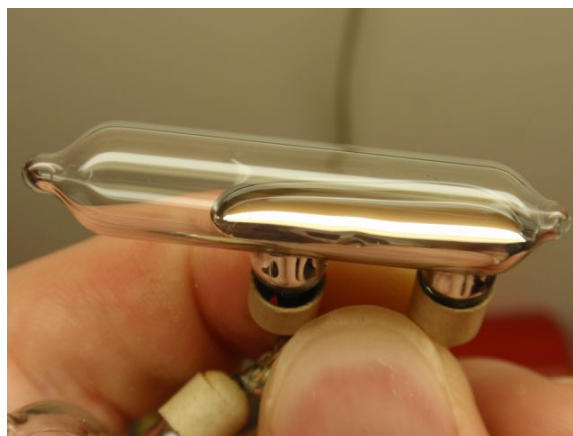


Obr. 2 Princip činnosti inklinometru

[12]

### 1.3 Rtuťové snímače

Rtuťový snímač je skleněná nádobka zpoloviny naplněná rtuťí. Zbytek nádoby je vyplněna vzduchem nebo vakuem. Při náklonu se rtuť přelívá vlivem gravitace, a tím zapojuje nebo rozpojuje obvod mezi elektrodami. Používají se například v relé, kde slouží jako eliminátory opotřebení běžných kontaktů. Dále našly uplatnění v prodejních automatech (detekce naklonění, otřesů, úderů), schodišťových automatech, stavebních stojích atd. Dnes jsou již pomalu nahrazovány akcelerometry kvůli toxicitě rtuti.



Obr. 3 Rtuťoměr

[13]

### 1.4 Akcelerometry

Akcelerometry jsou snímače, které primárně měří okamžité zrychlení (akceleraci) resp. jakoukoliv sílu, změnu síly působící na jeho měřící ústrojí. Pomoci akcelerometru můžeme měřit jak nárazovou sílu působící na snímač (dynamické zrychlení) tak stabilní gravitační sílu (statické zrychlení). Díky systémům snímání ve 3 osách se dají akcelerometry použít i jako snímače polohy, pozice a natočení.

Na trhu existuje mnoho firem nabízejících různé akcelerometry v různých provedeních např. STMicroelectronics, DELPHI, Kionix, ICSensors, Honeywell, VTI., Freescale..

Podle principu měření zrychlení se akcelerometry dělí na

- akcelerometry se seismickou hmotou
- tepelné akcelerometry
- MEMS kapacitní akcelerometry

Akcelerometry se seismickou hmotou se dále dělí na piezoelektrické akcelerometry a piezorezistivní akcelerometry. MEMS kapacitní se dají rozdělit podle toho v kolika osách měří zrychlení (jednoosé, dvouosé, tříosé). MEMS akcelerometry mají spoustu výhod jako je možnost měřit velmi nízké hodnoty g, nárazová odolnost až 10000g (termín Shock Survival) Další výhodou je nízké napájecí napětí (3V), malé rozměry, cena.

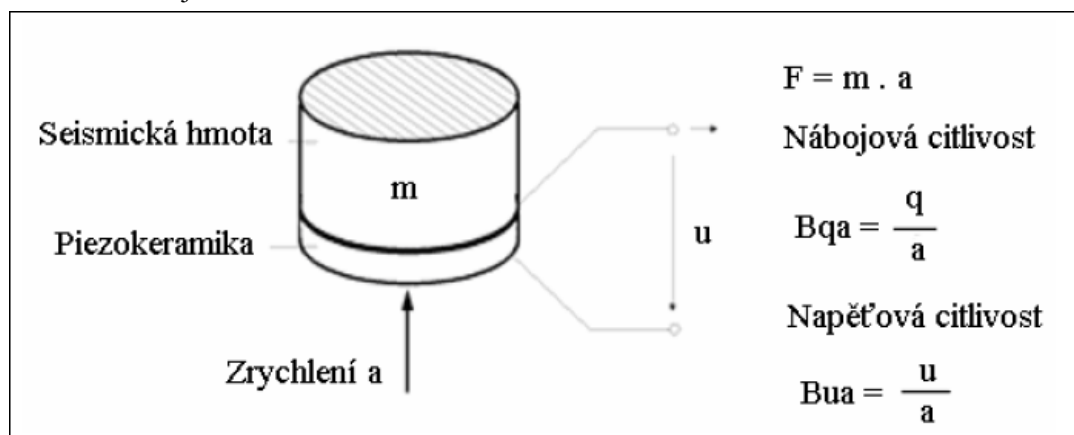
#### 1.4.1 Akcelerometry se seismickou hmotou

##### 1.4.1.1 Piezoelektrické akcelerometry

Princip činnosti piezoelektrických akcelerometrů spočívá v tlaku seismické hmoty na piezoelektrický krystal. Piezoelektrický krystal je z jedné strany připevněn k senzoru, z druhé strany je seismická hmota. Při působení síly na měřicí ústrojí akcelerometru dochází k mechanickému namáhání piezoelektrického krystalu vlivem tlaku seismické hmoty. Piezoelektrický krystal generuje elektrický náboj úměrný velikosti působící síly. Velikost působící síly je závislá na velikosti zrychlení  $a$  (konstantní) hmotnosti seismické hmoty, podle Newtonova zákona  $F=m \cdot a$ .

Vyprodukovaný elektrický náboj je následně zkonvertován vnitřní elektronikou akcelerometru na napěťový výstup.

Základní parametry akcelerometrů využívajících piezoelektrický materiál jsou tedy napěťová citlivost a nábojová citlivost.



Obr. 4 Zjednodušený princip činnosti piezoelektrického akcelerometru

Tyto akcelerometry se konstruují do 3 typů, podle způsobu použití.

- Smykový typ: akcelerometry tohoto typu dosahují lepších výsledků při parazitních vibracích pouzdra ( rezonanční frekvence...), jsou odolnější a pracují ve větším teplotním rozsahu
- Ohybový typ: specializovaná konstrukce pro měření seismické aktivity
- Kompresní typ: základní konstrukce pro průmyslové využití

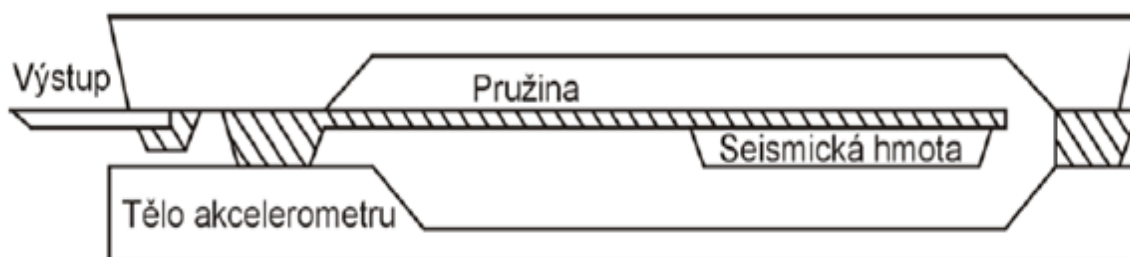
Výhodou těchto akcelometrů je snadná výroba, cena. Nevýhodou jsou větší rozměry a neschopnost měřit konstantní zrychlení (při konstantním tlaku seismické hmoty piezoelektrický materiál negeneruje el. náboj)

#### 1.4.1.2 Piezorezistivní akcelerometry

První piezorezistivní akcelerometr byl zkonstruován již v roce 1979, velmi dlouhou dobu byly a jsou stále nejpožívanější, pomalu je však vytlačují MEMS kapacitní akcelerometry.

Piezorezistivní akcelerometry pracují na podobném principu jako akcelerometry s piezoelektrickým krystalem. Hlavní rozdíl je v tom, že piezorezistivní akcelerometry nepoužívají piezoelektrický krystal ale piezorezistivní materiál, který mění svůj odpor při tlaku seismické hmoty. Piezorezistivní snímače jsou zapojeny do Wheatsonova můstku. Vlivem působící síly je piezorezistivní materiál (Si) deformován a mění tak svůj odpor.

Vnitřní konstrukci akcelometru si můžeme představit jako zátěž na piezorezistoru. Když působí síla na akcelerometr, zátěž setrvává v klidu dokud piezorezistor nepředá dostatek energie k pohybu. Síla působící na hmotu je potom přímo úměrná deformaci piezorezistoru.



Obr.5 Blokové schéma piezorezistivního akcelometru

[14]

Výhody piezorezistivních akcelometrů:

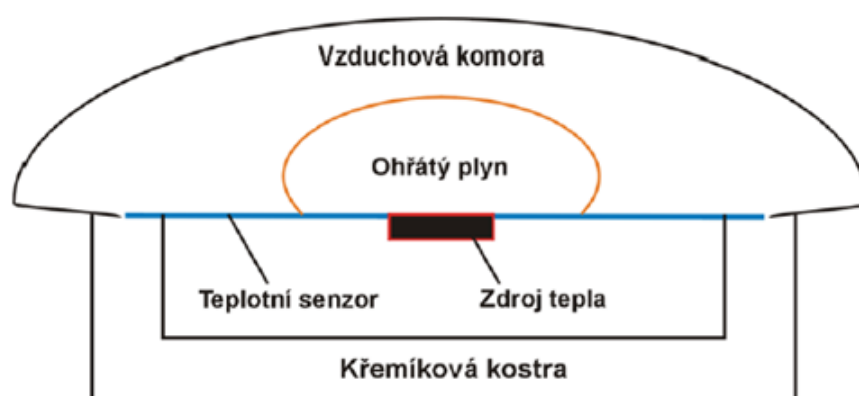
- velká citlivost měření (0,01g)
- možnost měřit konstantní zrychlení
- malá velikost, netečné pouzdro
- velká šířka pásma, lineární výstup
- nízká cena



Nevýhoda je citlivost na zrychlení v ostatních osách, požadujeme citlivost na zrychlení pouze v jednom směru. Tato citlivost se označuje jako křížová citlivost (Cross Sensitivity) a způsobuje chybu měření. Další nevýhodou je chyba měření způsobená teplotní roztažností. Tuto chybu je třeba kompenzovat samočinným testováním.

#### 1.4.2 Tepelné akcelerometry

Tepelné akcelerometry využívají setrvačnosti vzduchu v měřicí komůrce. Základem měřícího ústrojí je topné tělísko zahřívající vzduch v kulové měřicí komůrce na konstantní teplotu. Rozloženou teplotu v komůrce snímají teplotní senzory které jsou rovnoměrně rozloženy v komůrce. V klidovém stavu teplotní snímače snímají konstantní rozložení teploty v měřicí komůrce. Při pohybu akcelerometru se ohřátý vzduch v komůrce vlivem setrvačnosti pohybuje proti směru působené síly, kdežto komůrka je pevně fixována a pohybuje se ve směru působící síly (s akcelerometrem). Dojde tak ke změně teplotního rozložení v akcelerometru, na jedné straně komůrky teplota vzroste, na druhé klesne. Z teplotního rozdílu lze zjistit velikost zrychlení, a z nárustu nebo poklesu lze zjistit směr působící síly.



**Obr.6: Zjednodušené blokové schéma tepelného akcelerometru**

[14]

Nevýhodou tepelných akcelerometrů je náročnost na provozní podmínky.

### 1.4.3 MEMS kapacitní akcelerometry

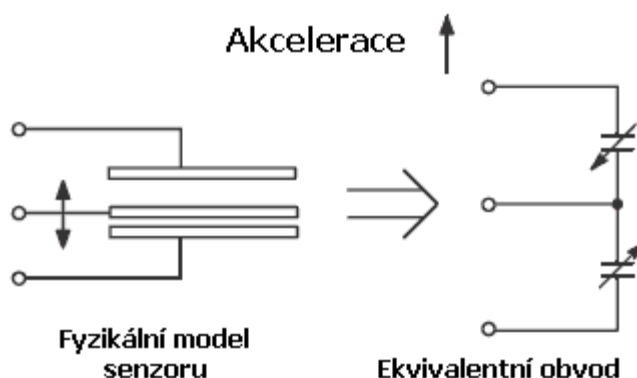
Produkty založené na MEMS technologii se taky někdy nazývají “inteligentní” produkty, jelikož tyto senzory obsahují jak mikromechanický měřicí systém tak i elektronický subsystém který dokáže naměřená data zpracovat.

MEMS akcelerometry (Micro-Electro-Mechanical Systems) se skládají z elektronických prvků a Mikromechanických částí. Elektronické prvky jako jsou řídicí a vyhodnocovací obvody, mechanické elementy, integrované obvody a akčních členy se vyrábějí technologiemi CMOS, BiCMOS. Mikromechanické části jsou zhotovovány technologií selektivního leptání kombinované s implementováním nových vrstev. To vše je možné osadit na křemíkový monokrystal díky moderním výrobním technologiím.

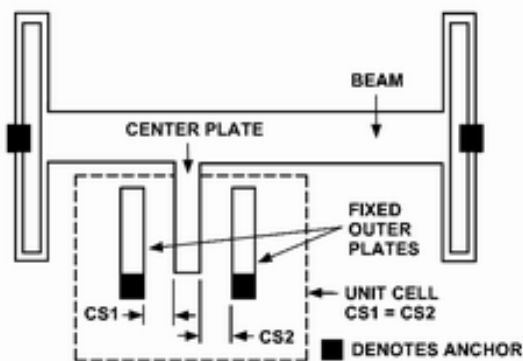
Hlavní výhodou technologie MEMS je výrazné zmenšení rozměrů a nízké napájecí napětí a nízká spotřeba snímačů založených na MEMS technologii. Další výhody jsou v kompaktnosti, odolnosti a nízké ceně snímačů.

MEMS technologie našlo uplatnění v mnoha oborech lidské činnosti, využívají se například v:

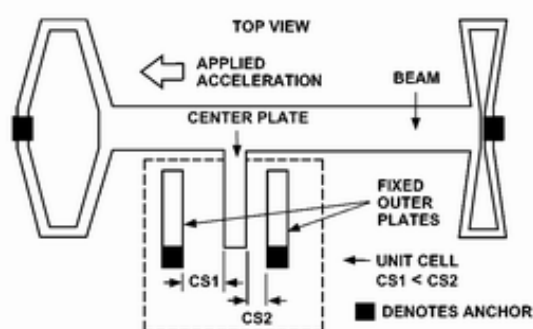
- Průmyslu: automobilový průmysl, ochranné systémy, detekce nárazu, robotika, stabilizace obrazu, monitoring seismické aktivity, kontrola opotřebení ložisek, kontrola rovnováhy, atd...
- Zdravotnictví: sportovní lékařství, ergonomické systémy, rehabilitační přístroje..
- Strojírenství a dalších průmyslových aplikacích.



Obr.7: Zjednodušený model principu fungování kapacitních akcelerometrů



Obr.8: Měřicí ústrojí v klidu



Obr.9: Měřicí ústrojí při působení zrychlení

[7]

Princip činnosti MEMS akcelerometru je založen na změně kapacity kondenzátoru umístěného v měřicím ústrojí. Působením síly (zrychlení) na měřicí ústrojí se změní kapacita kondenzátorů a ta je dále vyhodnocována a zpracována ve vyhodnocovacích obvodech akcelerometru. Vlastní mikromechanické měřicí ústrojí je tvořeno pružiny, nosníkem s pevnými úchyty. Pružiny umožní pohyb této struktury po povrchu monokrystalu. Při působení zrychlení se tato struktura deformuje, prohýbá a způsobí změnu kapacity kondenzátorů které jsou zapojené v diferenčním zapojení. Pohyblivá deska deformujícího se nosníku je umístěna mezi dvěma kondenzátory složených z pevných desek. Výstupem tohoto zapojení jsou 2 stejné obdélníkové signály stejné amplitudy, avšak vzájemně posunuté o  $180^\circ$ . Při působení zrychlení na senzor se prostřední deska posune od směru zrychlení, způsobí tak změnu dělicího poměru mezi kondenzátory. Zrychlení se zde převádí na amplitudu výstupu, směr působícího zrychlení charakterizuje fáze.

V dnešní době jsou MEMS akcelerometry hojně používané snímače, pomalu vytlačují starší akcelerometry založené na piezoelektrickém krystalu a piezorezistentním materiálu.

[1][3][7][12][13][14]

## 2 Bezdrátová komunikace

### 2.1 Technologie Zigbee

Komunikační technologie ZigBee je nový nízkorychlostní standard bezdrátové komunikace, zaměřený především na oblasti automatizace a řídicí techniky. Jedná se o bezdrátovou komunikační technologii schválenou jako mezinárodní standard nadnárodní organizací ZigBee Alliance a standardizační organizací IEEE. Tato perspektivní bezdrátová komunikační technologie najde uplatnění zejména v takových oborech, jako jsou řízení budov, dálkové ovládání, monitorování a diagnostika zařízení, vzdálené čtení měřených hodnot, počítačové periferie nebo spotřební elektronika.

### 2.2 Vznik ZigBee a jeho vlastnosti

Varianta	2.4 GHz	868 MHz	915 MHz
Frekvenční pásmo	ISM		ISM
Území	Celosvětově	Evropa	Amerika
Přenosová rychlost	250 kbps	20 kbps	40 kbps
Počet kanálů	16	1	10

Obr. 10 Rádiová pásma využívaná ZigBee

[5]

Standard ZigBee nemá působit jako přímá konkurence již zavedených komunikačních standardů typu Bluetooth, ale naopak jako jejich doplněk, který má rozšířit oblasti nasazení. ZigBee je vyvíjen mezinárodním konsorciem firem ZigBee Alliance a mezi hlavní účastníky jeho vývoje je možné jmenovat například nadnárodní firmy a korporace Freescale Semiconductor, Honeywell, Mitsubishi Electric, Motorola, Philips, Samsung, Invensys a další. Standard ZigBee je navržen jako jednoduchá bezdrátová komunikační síť a vyznačuje se takovými vlastnostmi, jako jsou jednoduchost, velmi nízká spotřeba energie, schopnost vytvářet statickou síťovou strukturu, spolehlivost a především příznivá cena.

### 2.3 Specifikace radiové části standardu

Standard ZigBee je založen na využití fyzické a linkové vrstvy podle mezinárodního standardu IEEE 802.15.4. Bylo pro něj definováno několik rádiových pásem, aby byl akceptovatelný v různých zemích s odlišnými předpisy a kritérii. Základním problémem při

definici rádiových pásem jsou především rozdíly v organizaci rádiových pásem v Americe a na evropském kontinentě. Aby se mohl standard uplatnit v obou těchto lokalitách, jsou pro něj definována tři rádiová pásma (obr. 17):

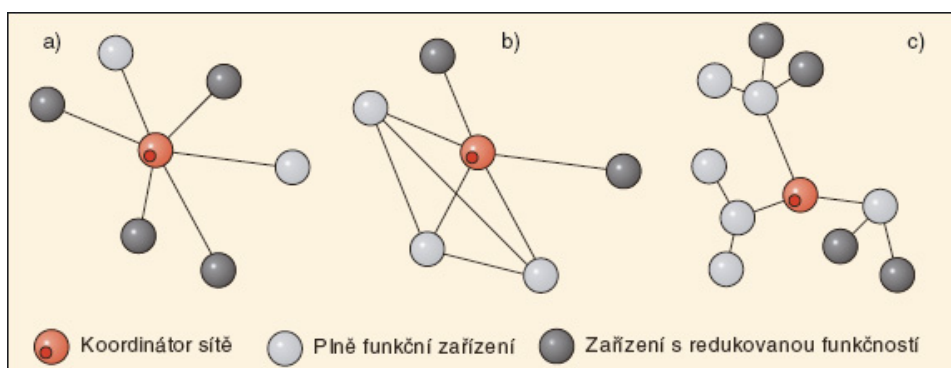
globální použití: pásmo ISM 2,4 GHz s 16 kanály a přenosovou rychlostí 250 kb/s;

Amerika a Austrálie: pásmo 915 MHz s 10 kanály a přenosovou rychlostí 40 kb/s;

Evropa: pásmo 868 MHz s jedním kanálem a přenosovou rychlostí 20 kb/s.

Technologie je založena na implementaci přístupové metody CSMA/CA k fyzickému médium, což znamená, že vlastní rádiová část standardu IEEE 802.15.4 této metody využívá na úrovni fyzické a linkové vrstvy komunikačního modelu. Vlastní standard IEEE 802.15.4 definuje komplexní komunikační protokol, který je založen na přenosu datových rámců. Jsou definovány čtyři typy komunikačních rámců využívané buď pro přenos užitečných datových informací, nebo k režijním účelům souvisejícím se sestavením, správou a řízením sítě:

Na základě časové synchronizace mezi centrální stanicí a koncovou stanicí dochází u “uspané” koncové stanice k probouzení ve vymezeném časovém intervalu, a poté jsou přeneseny užitečné informace.



Obr. 11 Topologie sítě ZigBee: typu hvězda (a), síť (b) a strom (c – nedoporučována)

[5]

## 2.4 Topologie sítě ZigBee

Standard IEEE 802.15.4 využívá pro adresaci jednotlivých zařízení binární adresovací kódy, které mohou být buď dlouhé (64 bitů), či zkrácené (16 bitů). Lokální adresa zkráceného adresovacího kódu umožňuje v jedné síti adresovat maximálně 65 535 zařízení. Každá sestavená síť je dále identifikována 16bitovým identifikátorem PAN ID, který slouží pro rozlišení překrývajících se sítí v případě, že v jednom prostoru dochází k vytvoření a sestavení více sítí standardu IEEE 802.15.4. Každou síť s unikátním PAN ID zakládá a spravuje koordinátor (centrální stanice), přičemž ostatní stanice pracují v módu koncové stanice. Každá koncová stanice

může být konfigurována pro funkci směrovače nebo koncového zařízení. Podle funkčnosti se zařízení dělí na plně funkční zařízení (FFD), která mohou zastávat funkci koordinátora nebo směrovače, a na redukováná zařízení RFD, která mohou fungovat pouze jako koncová zařízení.

Standard ZigBee založený na fyzické a linkové vrstvě IEEE 802.15.4 definuje tři typy síťové topologie (obr. 19). Základní je topologie typu hvězda (star topology), v níž je vždy definováno jedno zařízení, které přebírá funkci koordinátora sítě, a ostatní zařízení působí ve funkci koncových zařízení. V topologii typu strom (tree topology) slouží jedno zařízení jako koordinátor a ostatní jako koncová zařízení. Na rozdíl od topologie hvězda však nemusí všechna zařízení komunikovat přímo s koordinátorem, ale mohou využít jiné koncové zařízení v konfiguraci FFD ve funkci směrovače jako prostředníka. Díky tomu umožňuje uvedená konfigurace zvětšit vzdálenosti mezi koncovým zařízením a koordinátorem. Poslední definovanou topologií je topologie typu síť (mesh topology), která kombinuje vlastnosti topologií strom a hvězda (tzv. hybridní topologii strom a hvězda). Síťová topologie přináší největší funkčnost, protože umožňuje sestavit síť libovolným způsobem.

## 2.5 Spotřeba zařízení

Komunikační standard ZigBee je navržen pro aplikace, v nichž zařízení potřebují vysílat a přijímat pouze malé objemy dat a kde je vyžadována extrémně nízká spotřeba. Protokoly jsou proto navrženy s ohledem na co nejmenší spotřebu energie koncových zařízení, u kterých se předpokládá napájení z baterií. Koordinátor a směrovače by však neměly být napájeny bateriově, protože funkčnost sítě je na nich závislá.

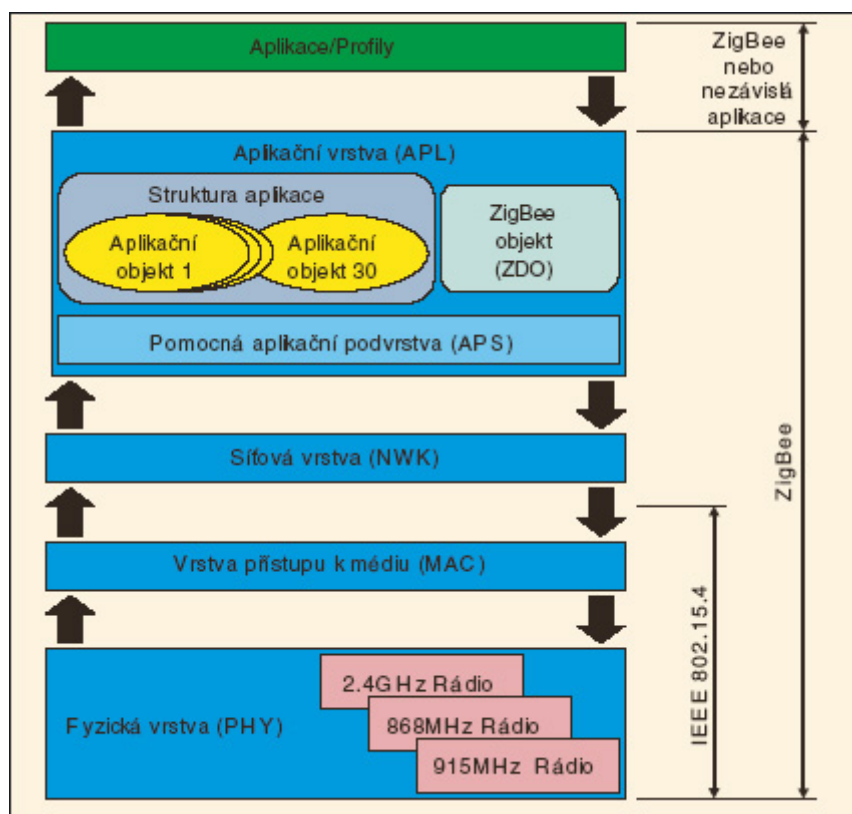
Při nejjednodušší topologii – hvězdě a využití technologie beacon se koncové zařízení aktivuje po přejmutí sekvence beacon a vyšle svá data. Koordinátor data přijme a uloží do paměti. Při přijetí další sekvence beacon indikuje koordinátor cílovému zařízení, že pro něj má data. Koordinátor data předá ve chvíli, kdy si je koncové zařízení vyžádá. Tento způsob zaručuje nejnižší spotřebu energie pro koncová zařízení, která jsou většinu doby přepnuta v úsporném režimu. Největší nároky jsou kladeny na koordinátora, který musí být schopen uložit všechna data pro jednotlivá zařízení.

## 2.6 Struktura protokolů ZigBee

Strukturu protokolů standardu ZigBee demonstruje obr. 20. Je zřejmé, že je navržena maximálně úsporně kvůli předpokládané implementaci do málo výkonných jednočipových 8bitových mikrokontrolérů s velmi omezenými paměťovými dispozicemi. Proto struktura protokolů nezabere více než asi 30 kB v systémové paměti, a je tedy mnohonásobně úspornější než standard Bluetooth, který vyžaduje více než 100 kB operační paměti.

Nad vrstvami standardu IEEE 802.15.4 je ve vlastním standardu Zigbee definována síťová vrstva (NWK) a struktura pro aplikační vrstvu (APL). Síťová vrstva provádí připojování k síti a

odpojování od ní, zabezpečení a směrování paketů. Dále zajišťuje objevování zařízení v rámci jednoho přeskočku. U koordinátora sítě ZigBee je odpovědná za start sítě a přiřazování adres nově začleněným zařízením.



Obr.12 Struktura protokolu ZigBee

[5]

Aplikační vrstva protokolu ZigBee se skládá z pomocné aplikační podvrstvy (APS), objektů ZigBee (ZDO) a uživatelských aplikačních objektů. Aplikační pomocná podvrstva udržuje párovací (binding) tabulky, které umožňují párování zařízení podle poskytovaných služeb a požadavků. Objekt ZigBee definuje roli zařízení v rámci sítě ZigBee (koordinátor, směrovač nebo koncové zařízení), zajišťuje objevování zařízení a vyhledávání poskytovaných služeb.

Komunikace mezi zařízeními sítě se odvíjí od profilu ZigBee, což je souhrn vlastností možných zařízení a definic typů a formátů zpráv mezi jednotlivými zařízeními tak, aby tvořily smysluplnou aplikaci. Profily ZigBee musí být jednoznačně určeny 16bitovým identifikátorem, který vydává na základě žádosti společnost ZigBee Alliance. V rámci profilu ZigBee si pak tvůrce aplikace definuje deskriptory zařízení, identifikátory zpráv a poskytované služby.

[5]

### 3 Návrh zařízení pro detekci převrnutí

V této kapitole popíšu strukturu řešení problému detekce a vizualizace převrnutí. Vše začíná měřením náklonu objektu a vyhodnocením jeho stavu natočení. Pro tento úkol se použil senzor zrychlení akcelerometr MMA7660. Vyhodnocena data z akcelerometru se musela nějakým způsobem zpracovat a bezdrátově odeslat. Pro moji práci se jevil vhodný ZigBee modem PAN4551 obsahující komplexní řešení ZigBee technologie. Z důvodu odlišného komunikačního rozhraní mezi akcelerometrem komunikujícím po sériové I2C a modulem PAN4551 používajícím standard UART bylo třeba najít převodník mezi těmito rozhraními. Vhodný byl jednoduchý, snadno ovladatelný a levný převodník SC18IM700 od firmy Philips. Převodník spolu s akcelerometrem a modulem PAN4551 jsou na měřicí desce. Měřicí deska komunikuje s okolím přes ZigBee standard, pro příjem informací se použila vývojová deska se stejnojmenným ZigBee modemem. Měřicí deska po zaznamenání alarmového stavu vyšle informaci přijímací stanici, ta po kontrole paketů odešle pakety na sériovou linku počítače. Přijímací deska komunikuje s počítačem pomocí sériové linky RS232.



Obr.13: Blokové schéma řešení bakalářské práce

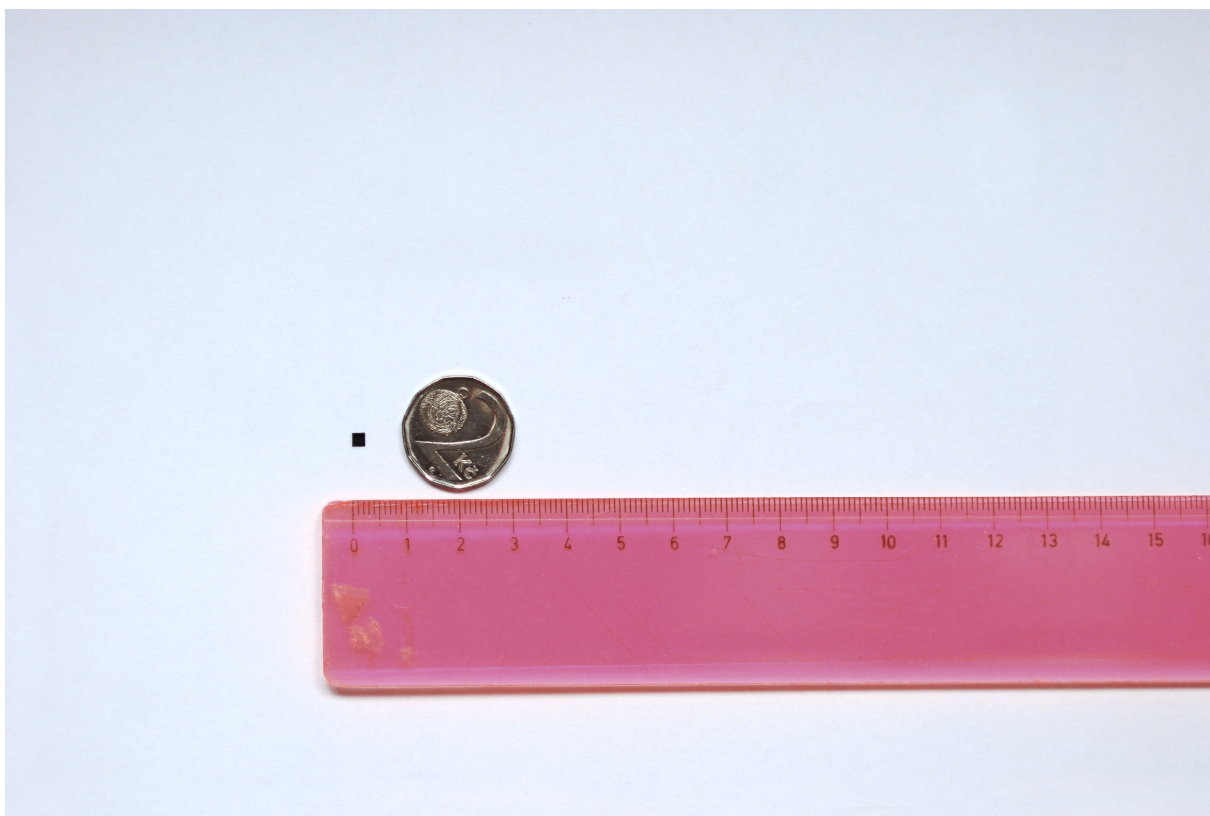
#### 3.1 MMA7660FC

Pro svoji práci jsem mohl použít kterýkoliv z senzorů náklonů, vybral jsem si však MEMS akcelerometry pro jejich zajímavost a možnosti programovatelnosti. Klíčovým měřicím senzorem tohoto systému je tedy akcelerometr, proto jsem věnoval značnou pozornost výběru tohoto senzoru. Při výběru akcelerometru bylo třeba zvažovat dostatečný měřicí rozsah a počet měřících os aby byl schopný detekovat převrnutí. Dále by měl mít dostatečnou vzorkovací frekvenci kvůli rychlosti detekce stavu převrnutí. Z důvodu dalšího zpracování dat v mikroprocesoru je vhodné komunikační rozhraní I2C a koneckonců byl požadavek na co nejnižší cenu měřicího modulu. Pro detekci převrnutí byl vhodný akcelerometr, který měří ve dvou osách, s minimální frekvencí vzorkování 1 vzorek za sekundu. Všechny parametry splňuje akcelerometr MMA7660FC.

MMA7660FC je digitální 6-bitový akcelerometr od firmy Freescale. Měří zrychlení ve třech osách. Je založený na MEMS technologii, měřicí systém a vyhodnocovací obvody jsou uloženy v malém DFN pouzdře o rozměrech 3mm x 3mm x 0,9mm. Vývody má ze 2 stran, na každé straně jich je 5, dohromady 10 vývodů. Disponuje digitálním I2C komunikačním rozhraním.



Akcelerometr umožňuje 3 módy provozu - Off mód, standby mód a active mód. Užitečná je funkce auto-wake/sleep mod kdy akcelerometr sám přejde ze standby modu do active modu. MMA7660FC dále nabízí funkce jako jsou: detekce volného pádu, otřesů, detekce stisku, detekce náklonu... Uživatel si může nastavit rychlost měření od 1 do 120 vzorků za sekundu.



**Obr.14: Velikost akcelerometru**

Velkou výhodou tohoto senzoru je vyhrazený registr TILT (0x03) . Senzor, pokud je v režimu active, kontinuálně ukládá informace o stavu natočení měřicího systému (Tilt Orientation Detection for Portrait/Landscape Capability) do tohoto registru. Tato funkce mi zjednodušila práci v tom, že nebylo třeba vymýšlet vlastní prahy, kdy dojde k převrnutí. Tato funkce je také jedním z hlavních důvodů proč jsem vybral tento senzor.

Mezi hlavní přednosti senzoru patří:

- měřicí rozsah  $\pm 1.5$  g
- nízkonapěťový provoz (1.71 V - 3.6 V )
- odolnost proti extrémním přetížením (až 10,000 g)
- možnost nastavení rychlosti měření (1 až 120 měření za sekundu)
- nízký odběr proudu: 0.4  $\mu$ A v off modu, 47  $\mu$ A v active modu
- komunikační rozhraní I2C
- provoz i v extrémních podmínkách ( $-40^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$ )
- detekce orientace akcelerometru (položený, na hraně, na výšku...)

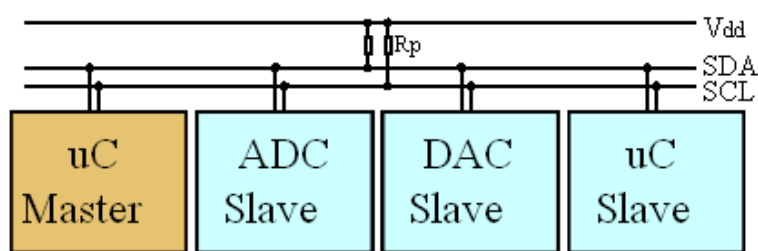
- úsporný režim provozu (standby mod)
- malé rozměry
- příznivá cena

### 3.1.1 Způsob komunikace

Akcelerometr MMA7660 komunikuje po digitálním I2C(Inter-Integrated Circuit) rozhraní, což je sběrnice vyvinutá firmou Philips v 80.letech. Hlavní cíl při vývoji I2C rozhraní bylo zjednodušení počtu vodičů, tím pádem i jednodušší výroba desek plošných spojů. Menší počet vodičů znamená taky odolnější přístroje vůči různým rušením ze strany jiných elektromagnetických přístrojů (elektromagnetická kompatibilita).

Komunikace po I2C probíhá po dvou vodičích, umožňuje datové propojení mezi jedním nebo obvykle více nadřazenými systémy (master) a podřízenou periférií (slave). Všechna připojená zařízení jsou na jedné sběrnici, při potřebě jsou volána svými unikátními adresami. I2C bylo navrženo pro jednoduché propojování integrovaných obvodů a jejich dalších rozšiřování. Je možno připojit obvody, které zvládají speciální protokol sběrnice. V dnešní době se rozhraní I2C hojně používá, například paměti RAM, EEPROM, A/D, D/A převodníky, obvody v televizní a audio technice, generátory hodinových pulsů...

Komunikace je zprostředkována dvěma linkami, těmi jsou sériová datová linka SDA (serial data) a linka pro hodinové signály SCL (serial clock). Data a adresy se jako v posuvných registrech posunují synchronně hodinovými impulsy. SDA i SCL linky lze použít pro obousměrnou komunikaci. Jsou vybaveny tzv. pull up odpory, mohou být staženy jakýmkoliv připojeným zařízením na nízkou úroveň.



**Obr.15 Základní schéma komunikace po I2C sběrnici**

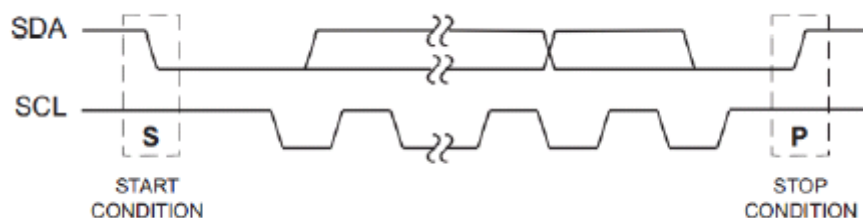
Data může vysílat jak master tak slave, neaktivní zařízení mají vysokou impedanci ale stále monitorují signály na sběrnici. Při použití jen jednoho master systému je zdrojem hodinového signálu toto zařízení. Maximální délka vodiče je dána jejich maximální kapacitou, a to 400 pF.

I2C používá následující systém komunikace:

- Při vysoké úrovni (HIGH) vodičů SCL i SDA je komunikace neaktivní a systém je v klidovém stavu
- Podmínka pro START: linka SDA přejde na nízkou úroveň (LOW), linka SCL zůstává na vysoké úrovni
- Podmínka Stop: SCL zůstává HIGH, SDA přejde z LOW úrovně do HIGH.
- Při přenosu dat vysílač přivodí na SDA osm bitů, které jsou hodinovými pulsy na lince SCL posouvány dále. První bit má nejvyšší váhu.
- Acknowledge (potvrzení) : koncové zařízení potvrdí příjem osmi bytu tak, že stáhne SDA na nízkou úroveň dokud master nepošle další hodinový signál na SCL. Acknowledge taky znamená, že bude přijímat další byte. Při nepotvrzení znamená, že jde od ukončení přenosu.

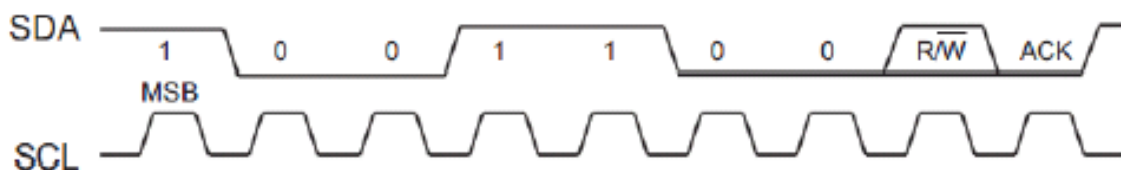
Při přenosu dat zařízení master vyrobí podmínku pro START, následuje adresa součástky přenesená v bitech 7 až 1, poslední bit 0 charakterizuje směr přenosu dat. Zařízení slave vyšle ACK (potvrzení). Poté následuje vyslání datových byte masterem, slave vytvoří ACK a může přijímat další byte nebo master ukončí spojení buď neexistujícím potvrzením nebo podmínkou STOP.

[7]



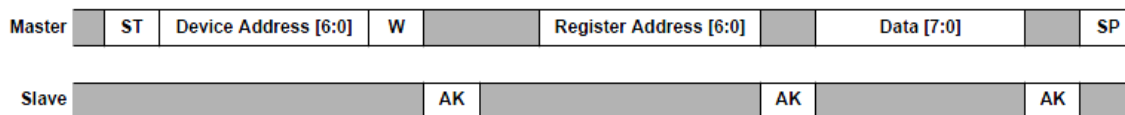
Obr.16: Start, Stop podmínky

Pro čtení dat od periferie je třeba s adresou zařízení poslat poslední bit 0 (R/W bit). Hodnota 1 znamená čtení (R), 0 je pro zápis (W). Master vyšle osm CLK impulsů, přijme osm datových bitů z periferie. Pro příjem dalších bitů musí poslat devátý hodinový impuls. Každá součástka komunikující po rozhraní má svoji adresu danou od výrobce. Maximální frekvence hodinových impulsů pro I2C je většinou v integrovaných obvodech do 100kHz.



Obr.17 Adresa akcelerometru MMA7660FC, orientace dat, potvrzení

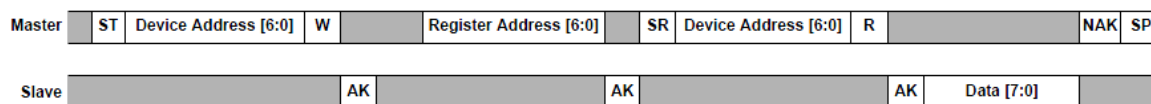
[7]



Obr.18 Schéma pro zápis bytu

[7]

Pro zápis více dat se po ACKNOWLEDGE posílají další data, podle potřeby zase dojde k potvrzení a tím je umožněno posílat další.

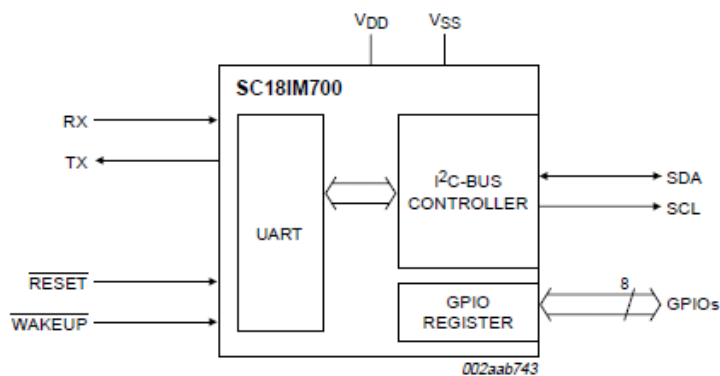


Obr.19 Schéma pro čtení

[4][7]

### 3.2 SC18IM700

Modul SC18IM700 v mé práci hraje úlohu překládače mezi dvěma komunikačními rozhraními. Převodník obsluhuje komunikační standardy UART mikrokontrolérů a I2C standard. Na jedné straně je akcelerometr používající rozhraní I2C, na straně druhé je UART, používá jej mikroprocesor užitý v mé práci. SC18IM700 může fungovat jako řídicí jednotka I2C komunikace, řídí arbitrace, časování a specifické sekvence I2C protokolů. Převodník disponuje vlastními registry které lze číst a přepisovat pomocí ASCII zpráv.



Obr.20: Blokový diagram SC18IM700

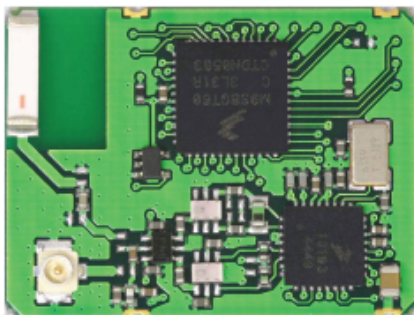
Některé vlastnosti převodníku:

- obsluha standardu I2C, UART
- 8 programovatelných I/O pinů
- vysoká rychlost UART 460.8 kbit/s a I2C 400 kbit/s
- napájecí napětí 2.3 – 3.6V
- úsporné režimy

Pro moji bakalářskou práci jsou důležité komunikační linky Rx, Tx, SDA, SCL. Vývody SDA, SCL jsou komunikační I2C, jsou připojeny na stejnojmenné vývody akcelerometru. Rx, Tx jsou standardu UART, jsou zkříženě připojené k modulu PAN4551. Schéma zapojení je v příloze. Piny GPIO, GPIO1 až GPIO5 jsou programovatelné piny, zůstávají nezapojené.

### 3.3 PAN4551

Modul PAN 4551 je hybridní čip firmy Panasonic, je postaven na referenčním návrhu firmy Freescale Semiconductor a využívá součástkové základny téhož výrobce. Základní částí hybridního čipu PAN4551 je 8bitový mikrokontrolér s označením MC9S08GT60 a rádiový čip MC13192 kompatibilní se standardem IEEE 802.15.4. Použitím prvků s nízkou spotřebou se průměrná spotřeba modulu pohybuje kolem 0,1 mW.



Obr.21: Modul PAN4551

[15]

Vlastnosti modulu :

- komunikuje na frekvenci 2.4GHz
- 3 možnosti připojení antény: vestavěná, připojitelná (50Ω), keramická
- 16 volitelných komunikačních kanálů, maximální rychlost 250 kbps
- možnost úsporných režimů
- nízké napájecí napětí (2.1V – 3.4V), optimum 2.7V
- malá velikost (20.0mm x 26.5mm x 3.0mm)
- 60k flash a 4k RAM paměť
- 5ti kanálový A/D převodník

### 3.3.1 Mikroprocesor MC9S08GT60

Mikroprocesor užitý v mé práci patří do řady mikroprocesorů s označením HCS08. Jsou vyráběny moderním výrobním procesem na bázi 0.25μ technologie. Výsledkem použití této technologie jsou vysoce integrované a mnohostranné součástky, které lze použít v širokém rozsahu řídicích aplikací pro průmyslové použití a pro řízení bílé techniky. Mezi hlavní přednosti těchto mikrořadičů patří vysoký procesní výkon (až 40MHz), napájecí napětí (2.7 až 5.5 V), nízká spotřeba energie ve všech stavech (provozní, klidový) a vysoká integrace periférií na čipu. Tento mikroprocesor se v mé práci programuje ve vývojovém prostředí Freescale Code Warrior, v jazyce C.

Některé vlastnosti mikroprocesorů řady HCS08:

- pracovní frekvence CPU 40MHz
- HC08 instrukční sada s podporou BGND instrukce
- Integrované ladicí rozhraní
- podpora do 32 zdrojů přerušení (interrupt/reset sources)
- úsporné režimy

Vlastnosti MCU MC9S08GT60:

- 60k flash paměti, 4k RAM
- 8mi kanálový 10bit AD převodník
- 2 komunikační rozhraní SCI (seriál)
- sériové komunikační rozhraní s periférií - SPI (Seriál peripheral interface)
- možnost více zdrojů clock: interní generátor hodin, krystalový generátor, externí hodiny
- 3 kanálová a 5 kanálová 16 bitová pulsně šířková modulace

### 3.4 Vývojová deska EvalKit PAN4551

Pro vývoj aplikací je nutné mít k dispozici univerzální kit s vývojovým softwarem. Deska disponuje čtyřmi propojenými sockety. 3 slouží k připojení periferie a senzorů, jeden socket je pro modul PAN4551. Dále je na desce 5 programovatelných tlačítek, 4 diody. Napájí se ze zdroje 7-9V DC nebo přes USB rozhraní. Je připojitelný BDM programátor USB HCS08/HCS12 Multilink, dodávaný v setu. Komunikační rozhraní s PC je řešeno po sériové lince RS232.

Základní vlastnosti:

- USB programovací a ladící rozhraní
- 9pin D-SUB RS232 pro uživatelskou aplikaci
- Koax. napájecí konektor pro 7-9V
- BDM vývod
- 4x LED připojené k vývodům MCU
- 1x tlačítko RESET
- 5x uživatelská tlačítka



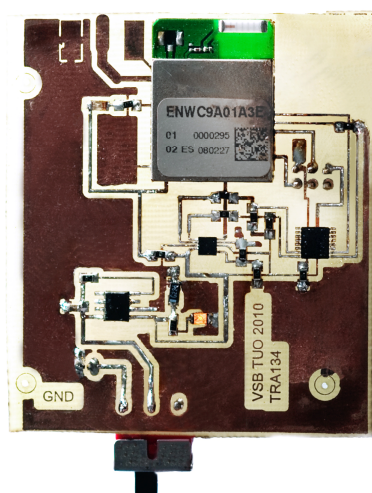
Obr.22: EvalKit s modulem PAN4551



## 4 Realizace systému

Byla zhotovena deska plošných spojů pro detekci a odeslání informace o převrnutí. Návrh desky se dělal v programu Eagle Layout Editor 5.7.0. Deska je dělána tak, aby byla co nejmenších rozměrů. Je napájena 9V baterií, v zapojení je lineární stabilizátor LE33CV který zásobuje akcelerometr MMA7660, převodník SC18IM700 a modul PAN4551 napětím 3.3V. Pro správnou funkci bylo třeba přidat kondenzátory o kapacitách 2m2 a 100n mezi zemí a vstupním i výstupním napětím. Dále bylo třeba 7 odporových propojek o hodnotě 0Ω. Pro signalizaci napájení byla použita oranžová LED dioda. Další 2 tzv. pull up odpory 4.7K jsou nutné pro správnou funkci komunikace I2C. Na desce je tlačítko typu switch pro pohodlné zapínání zařízení.

Modul PAN4551 má piny zespodu takže se musel připéct v horkovzdušné peci. Pec se nastavila na ohřev ve 2 intervalech, v prvním intervalu se nastavila teplota 150°C po dobu 90s, následovalo ohřátí na 235°C po dobu 10s. Ostatní součástky včetně akcelerometru se připájely mikropájkou. Elektrické schéma vytvořené desky plošných spojů se nachází v příloze.



Obr.23: Deska plošných spojů

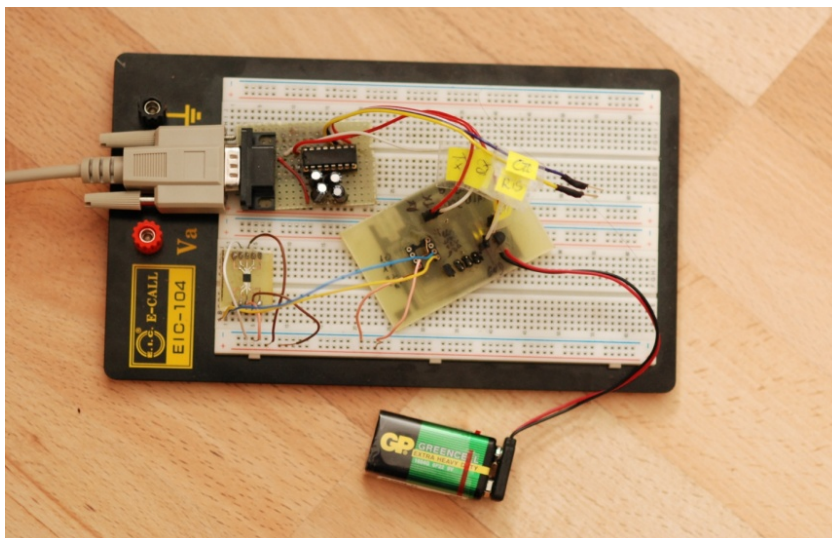
### 4.1 Zhotovení testovacího modulu

Pro otestování a pochopení funkce akcelerometru byla vytvořena deska plošných spojů pro akcelerometr. Deska je rozměrů 3x4cm, má vyvedené všechny piny akcelerometru.

Protože akcelerometr komunikuje po rozhraní I2C, bylo třeba převodních modulů SC18IM700 a MAX232 pro komunikaci s počítačem. Na testování se použilo nepájivé pole, do kterého se vložil akcelerometr, zapojily se vývody SCL, SDA z akcelerometru do modulu s SC18IM700. Z modulu SC18IM700 se vyvedly výstupy pro napájení akcelerometru VDD +3.3V



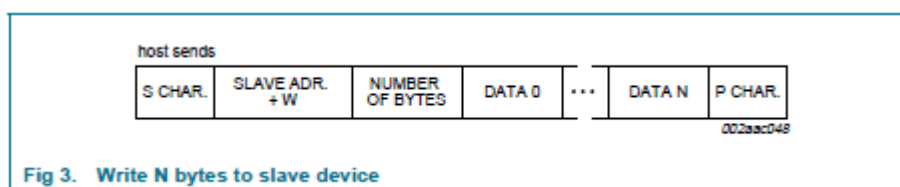
a GND. Pro zobrazení výsledku na PC se použil převodník MAX232 kvůli převodu z UART (Rx, Tx) výstupu z SC18IM700 na RS232 .



Obr.24: Zapojení testovacích modulů

## 4.2 Popis komunikačního protokolu

Pro testování akcelerometru bylo nutno v programu RealTerm v záložce Port vybrat příslušný COM port, nastavit Baud na 9600. Dále v záložce display nastavit Display As Hex pro snadné čtení komunikace.



Obr.25: Zápis bytu do zařízení

[11]



Figure 12. Single Byte Write

Obr.26: Zápis bytu do registru akcelerometru

[7]

Table 3. ASCII commands supported by SC18IM700

ASCII command	Hex value	Command function
S	0x53	I <sup>2</sup> C-bus START
P	0x50	I <sup>2</sup> C-bus STOP
R	0x52	read SC18IM700 internal register
W	0x57	write to SC18IM700 internal register
I	0x49	read GPIO port
O	0x4F	write to GPIO port
Z	0x5A	power down

Obr.27: Tabulka ASCII kódu

[11]

Nejprve bylo potřeba zapnout akcelerometr. Pro zapisování se запиše do záložky Send příkaz ve tvaru:

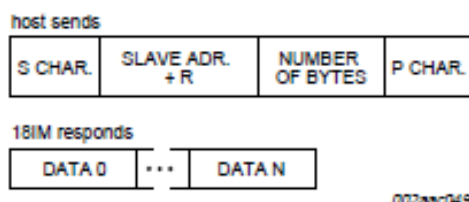
**Start (0x53) | adresa akcelerometru + W (0x98) | počet bytu | registr | hodnota | stop (0x50)**

Po příkazu start je následována výchozí adresa akcelerometru nastavená výrobcem 1001100 (0x4C) + hodnota 0 pro zápis nebo 1 pro čtení. Pro zápis bude tedy druhá hodnota 0x98, pro čtení 0x99. Dále příkaz pokračuje informací o počtu datových bytu, registrem do kterého chceme zapisovat a hodnota, kterou chceme zapsat. Následuje vyslání datových byte masterem, slave vytvoří ACK a může přijímat další byte nebo master ukončí spojení buď neexistujícím potvrzením nebo podmínkou STOP (0x53).

Pro čtení dat od periferie je třeba s adresou zařízení poslat poslední bit 0 (R/W bit). Hodnota 1 znamená čtení (R) , 0 je pro zápis (W). Master vyšle osm CLK impulsů, přijme osm datových bitů z periferie. Pro příjem dalších bitů musí poslat devátý hodinový impuls.

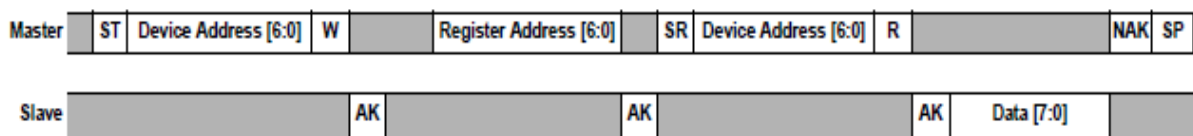
Sekvence pro čtení bytu z akcelerometru:

**Start (0x53) | adresa akcelerometru + R | počet bytu | registr | Start (0x53) | adresa akcelerometru + W | počet bytu | stop (0x50)**



Obr.28: Čtení bytu

[11]



Obr.29: Posloupnost příkazu pro čtení z akcelerometru

[7]

Pro obsluhu komunikace v PC jsem zvolil program Realterm. Dokáže zachytávat, kontrolovat, anebo ladit (vysílat) data na sériovém portu. Je to program, který je pro ovládání sériového portu daleko lepší než Hyperterminal. Nemá ale podporu pro vytáčení telefonních čísel což v mojí práci není potřeba.

Program umožňuje:

- Zobrazovat přijímaná data přes port
- Vysílat sekvence dat (čísla nebo celý soubor)
- Zachytávat přijímaná data (určitou dobu, nebo počet)
- Zapisovat přijímaná data do souboru
- Invertovat obě linky (Rxd, Txd)
- Nastavit maskování přijatých bytů pomocí log. opeací XOR, AND
- Ovládání nejen z přík. řádky nebo myši, ale i pomocí ActiveX

V záložce Display lze vybrat v jakém formátu se budou zobrazovat data v hlavním okně jež je umístěno nad záložkami. Dále se zde dá vybrat jak velký datový rámec se má zobrazovat. V záložce Port se nastavuje číslo portu, datová rychlost, typ parity, počet datových bitů a počet stop bitů. Ve třetí záložce Capture se nastavuje cesta k souboru do kterého se zapisují příchozí data. Konec zapisování je možno nastavit podle času, anebo podle počtu přijatých bytů. Je možné vypnout zobrazování příchozích dat, kvůli nižšímu zatěžování procesoru.

### 4.3 Obsluha akcelerometru

Akcelerometr musíme jako všechny jiné elektronické zařízení před používáním zapnout.

Pro zapnutí akcelerometru do režimu active (měření) je třeba z programu Realterm poslat příkaz v podobě hexadecimálních čísel ve tvaru

0x53	0x98	0x02	0x07	0x01	0x50
Start I2C komunikace	Adresa akcelerometru + směr přenosu dat (W)	Počet přenes. bytů	Cílový registr	Zapisovaná hodnota	Stop I2C komunikace

tímto příkazem se zapíše hodnota 01 do registru 07.

Pro kontrolu můžeme načíst hodnotu z registru 07 příkazem

<b>0x53</b>	<b>0x98</b>	<b>0x01</b>	<b>0x07</b>	<b>0x53</b>	<b>0x99</b>	<b>0x01</b>	<b>0x50</b>
Start I2C komunikace	Adresa akcelerometru + W	Počet přenes. bytů	Cílový registr	Opakovaný start	Adresa akcelerometru + R	Počet bytů	Stop I2C komunikace

\$07: Mode Register (Read/Write)

MODE

D7	D6	D5	D4	D3	D2	D1	D0
IAH	IPP	SCPS	ASE	AWE	TON	-	MODE
0	0	0	0	0	0	0	0

NOTE: Writing to the Mode register resets sleep timing, and clears the XOUT, YOUT, ZOUT, TILT registers. Reading to the Mode register resets sleep timing.

NOTE: The device must be placed in Standby Mode to change the value of the registers.

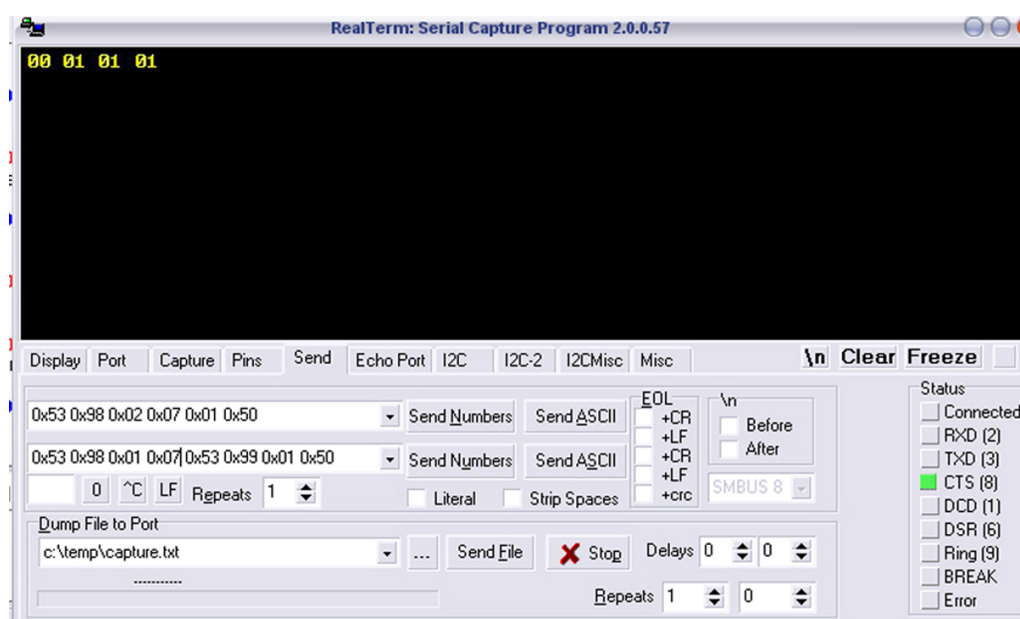
Table 10. Modes

Mode of Operation	D0 - MODE	D2 - TON
Standby Mode	0	0
Test Mode	0	1
Active Mode	1	0

Obr.30: Mode Registr

[7]

V okně programu Realterm se vypíše hodnota registru 07 po zapsání hodnoty 01 (zapnutí).

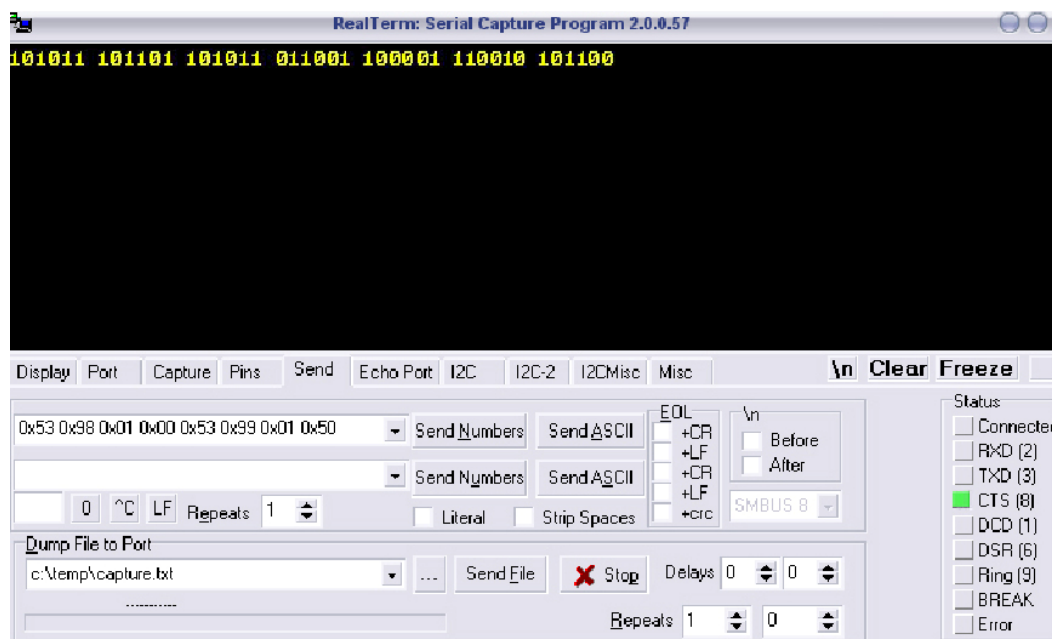


Obr. 31: Program Realterm, čtení z registru 07

Funkčnost měření jsem otestoval příkazem pro čtení z registru 0x00 – náklon x-ové osy. Různě jsem natáčel akcelerometr pro získání zajímavějších údajů.

Příkaz pro čtení z registru 0x00:

<b>0x53</b>	<b>0x98</b>	<b>0x01</b>	<b>0x00</b>	<b>0x53</b>	<b>0x99</b>	<b>0x01</b>	<b>0x50</b>
Start I2C komunikace	Adresa akcelerometru + W	Počet přenes. bytů	Cílový registr 0x00	Opakovaný start	Adresa akcelerometru + R	Počet bytů	Stop I2C komunikace



Obr. 32: Program Realterm, čtení z registru 00 (náklon X-ové osy)

Z datasheetu akcelerometru na straně 29 lze vyčíst, že například hodnota 101011 znamená náklon v X-ové ose  $-79.86^\circ$ , hodnota g =  $-0.984g$ , náklon v Z-ové ose  $-10.14^\circ$ .

Podobně se dá vyčíst hodnota z ostatních os.

Address	Name	Definition	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$00	XOUT	6-bit output value X	-	Alert	XOUT[5]	XOUT[4]	XOUT[3]	XOUT[2]	XOUT[1]	XOUT[0]
\$01	YOUT	6-bit output value Y	-	Alert	YOUT[5]	YOUT[4]	YOUT[3]	YOUT[2]	YOUT[1]	YOUT[0]
\$02	ZOUT	6-bit output value Z	-	Alert	ZOUT[5]	ZOUT[4]	ZOUT[3]	ZOUT[2]	ZOUT[1]	ZOUT[0]
\$03	TILT	Tilt Status	Shake	Alert	Tap	PoLa[2]	PoLa[1]	PoLa[0]	BaFro[1]	BaFro[0]
\$04	SRST	Sampling Rate Status	0	0	0	0	0	0	AWSRS	AMSRS
\$05	SPCNT	Sleep Count	SC[7]	SC[6]	SC[5]	SC[4]	SC[3]	SC[2]	SC[1]	SC[0]
\$06	INTSU	Interrupt Setup	SHINTX	SHINTY	SHINTZ	GINT	ASINT	PDINT	PLINT	FBINT
\$07	MODE	Mode	IAH	IPP	SCPS	ASE	AWE	TON	-	MODE
\$08	SR	Auto-Wake/Sleep and Portrait/Landscape samples per seconds and Debounce Filter	FILT[2]	FILT[1]	FILT[0]	AWSR[1]	AWSR[0]	AMSR[2]	AMSR[1]	AMSR[0]
\$09	PDET	Tap Detection	ZDA	YDA	XDA	PDTH[4]	PDTH[3]	PDTH[2]	PDTH[1]	PDTH[0]
\$0A	PD	Tap Debounce Count	PD[7]	PD[6]	PD[5]	PD[4]	PD[3]	PD[2]	PD[1]	PD[0]
\$0B-\$1F	Factory	Reserved	-	-	-	-	-	-	-	-

Obr.33: Přehled registrů

V mojí práci používám registr \$03 TILT. Řešení detekce převrnutí je z důvodu zjednodušení práce založeno na čtení z registru TILT. Tento registr je jen pro čtení, do tohoto registru se ukládají informace o aktuálním stavu polohy akcelerometru. Registr dále umožňuje detekci poklepání (Tap) a detekci otřesů (shake).

#### \$03: Tilt Status (Read only)

##### TILT

D7	D6	D5	D4	D3	D2	D1	D0
Shake	Alert	Tap	PoLa[2]	PoLa[1]	PoLa[0]	BaFro[1]	BaFro[0]
0	0	0	0	0	0	0	0

##### BaFro[1:0]

00: Unknown condition of front or back  
 01: Front: Equipment is lying on its front  
 10: Back: Equipment is lying on its back

##### PoLa[2:0]

000: Unknown condition of up or down or left or right  
 001: Left: Equipment is in landscape mode to the left  
 010: Right: Equipment is in landscape mode to the right  
 101: Down: Equipment standing vertically in inverted orientation  
 110: Up: Equipment standing vertically in normal orientation

##### Tap

1: Equipment has detected a tap  
 0: Equipment has not detected a tap

##### Alert

0: Register data is valid  
 1: The register was read at the same time as MMA7660FC was attempting to update the contents. Re-read the register

##### Shake

0: Equipment is not experiencing shake in one or more of the axes enabled by SHINTX, SHINTY, and SHINTZ  
 1: Equipment is experiencing shake in one or more of the axes enabled by SHINTX, SHINTY, and SHINTZ

Obr.34: TILT registr

K řešení mojí práce používám sekvenci pro uvedení akcelerometru do provozu a následně sekvenci příkazu pro čtení z registru TILT:

0x53	0x98	0x01	0x03	0x53	0x99	0x01	0x50
Start I2C komunikace	Adresa akcelerometru + W	Počet bytů	Cílový registr (TILT)	Opakovaný start	Adresa akcelerometru + R	Počet bytů	Stop I2C komunikace

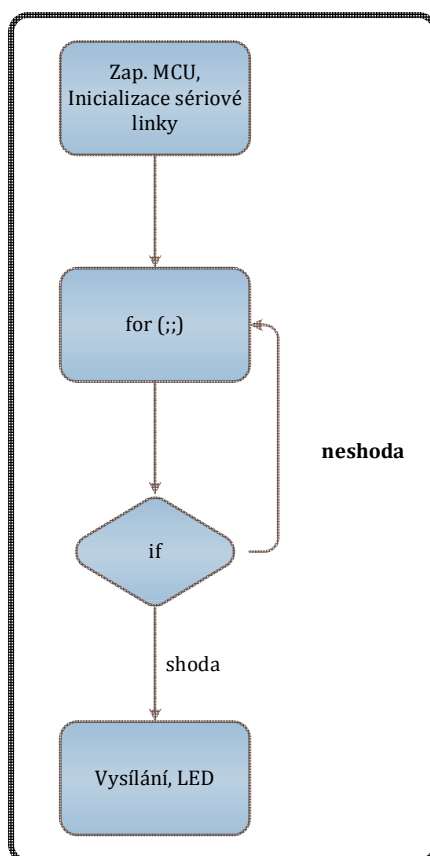
Akcelerometr odpovídá na příkaz čtení, v tabulce jsou možnosti a hodnoty které akcelerometr posílá ve stavech při převrnutí. Vyčtené informace se dále vyhodnocují v mikroprocesoru.

Natočení	Data
Levý bok	110
Pravý bok	1010
Vertikální poloha	11010
Invertní vertikální	10110

## 4.4 Program pro MCU

Pro vyhodnocení stavu převrnutí bylo třeba vytvořit program pro čtení dat z akcelerometru a jejich následné vyhodnocení. Program se vyvíjel v jazyce C ve vývojovém prostředí Code Warrior a testoval na vývojové desce EvalKit PAN4551 s procesorem MC9S08GT60 řady HCS08.

Program na měřicím zařízení je založený na neustálém čtení registru TILT. Po inicializaci sériové linky je akcelerometr přepnut do režimu active. Poté v nekonečné smyčce for mikroprocesor vyčítá informace z registru TILT a porovnává s alarmujícími stavy (if). Tyto stavy byly popsány v předchozí kapitole.



**Obr.33: Vývojový diagram**

Jakmile nastane shoda se zmíněnými stavy, mikroprocesor vyhodnotí alarm. Vyšle zprávu přijímací stanici a začne vizualizovat alarm rozsvícením signální diody přímo na měřicím zařízení.

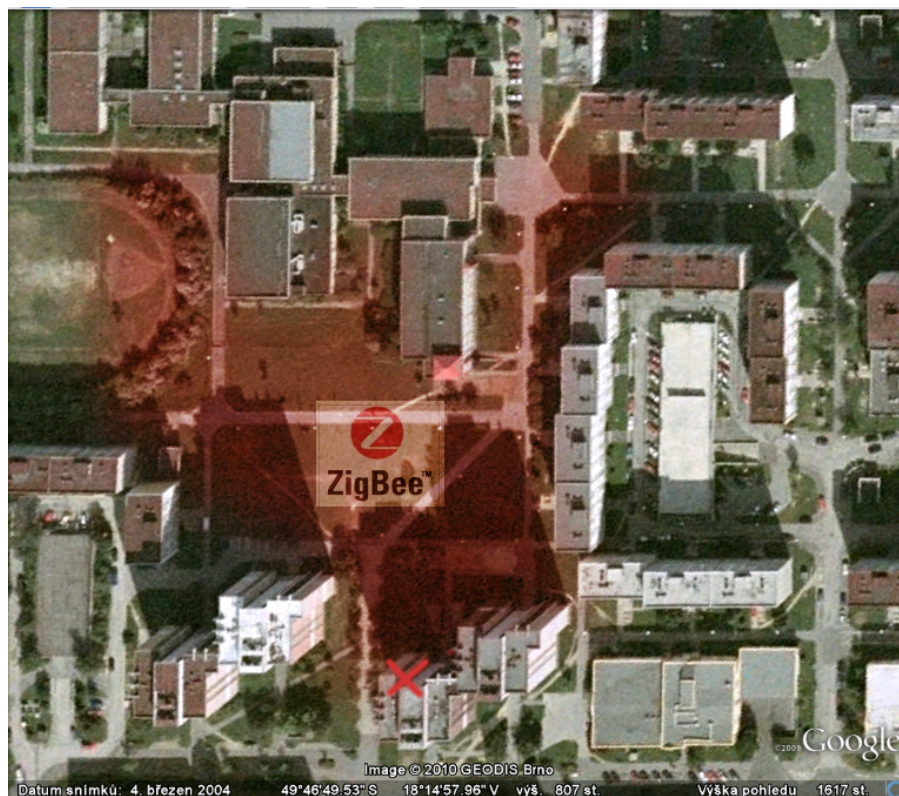
Na přijímací stanici proběhne kontrola přijatých paketů. Jestliže bude první paket 0x02, čtvrtý 0x80 a pátý 0x03, MCU vyšle zprávu na sériovou linku počítače.

Zdrojové kódy a vývojové diagramy programů pro vysílač a přijímač jsou v příloze.



## 4.5 Dosah zařízení

Při testování se zjistilo, že zařízení komunikuje přes ZigBee standard bez problémů do 165m za podmínky, že zařízení a přijímač jsou v přímé viditelnosti. Vliv na dosah může mít stav baterie nebo okolní rušení. Problémy má s překážkami ve formě zdí a hustých listí. V bytě komunikuje do 13m přes 3 zdi.



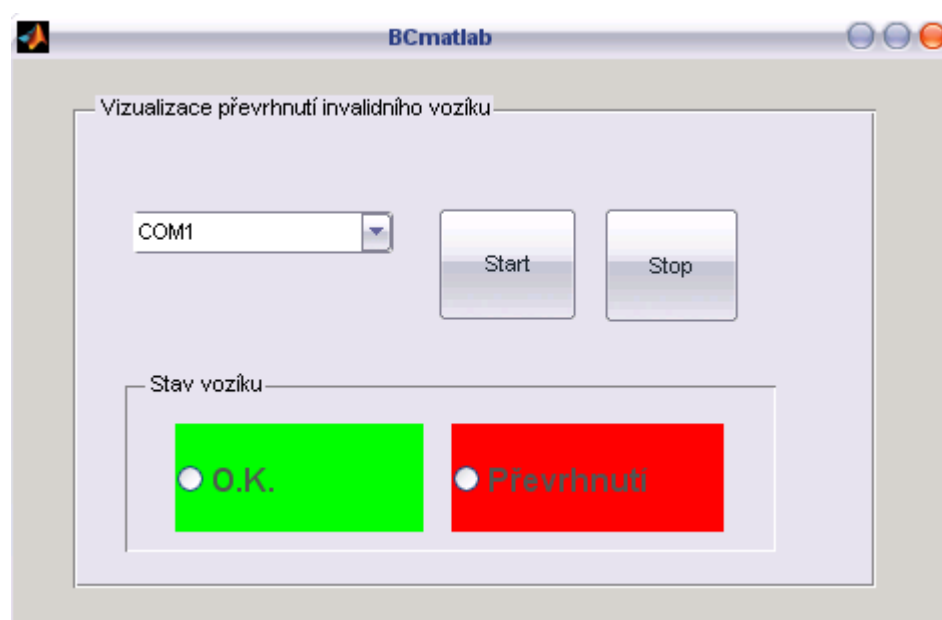
**Obr.33: Dosah zařízení**



## 5 Vizualizace převrnutí

Po detekci převrnutí a odeslání zprávy na sériovou linku je třeba srozumitelně sdělit uživateli, že nastal alarm. Program pro vizualizaci byl vytvořen v programu MATLAB, přes GUI prostředí. Bylo použito popup menu pro výběr sériových portů a 2 radiobutton k signalizaci stavů. Ohled byl brán na jednoduchost ovládání a přehlednost vizualizace.

V hlavním okně je na výběr číslo sériového portu, tlačítka Start, Stop. Po stisknutí tlačítka Start začne program sledovat vybranou sériovou linku. Při převrnutí program jasně sdělí stav zařízení. Datum a čas převrnutí se automaticky uloží do textového souboru. Zdrojový kód programu je v příloze.



Obr.34: Okno vizualizačního programu

## Závěr

Mým cílem bylo zrealizovat systém pro automatickou detekci převrnutí invalidního vozíku. Práce je založena na senzoru zrychlení, přenos dat ze senzoru k přijímací desce se provádí přes bezdrátovou technologii ZigBee. Zpracování dat je řešeno mikroprocesorem.

Důležitá, ne-li nejdůležitější část práce byl správný výběr senzoru pro vyhodnocení převrnutí. Mnou vybraný akcelerometr se ukázal jako naprosto vhodný pro tuto práci díky registru TILT. V práci jsem se seznámil s procesem výroby desky plošných spojů od návrhu schématu v programu Eagle po leptání, pájení a osazování součástek. Abych mohl plně využít možností akcelerometru, musel jsem porozumět komunikaci akcelerometru s převodníkem a mikroprocesorem.

Fáze testování a ověření funkčnosti akcelerometru s komunikačními moduly SC18IM700 a MAX232 přes sériovou linku počítače proběhla úspěšně. Akcelerometr po zapnutí měřil velikost zrychlení, reagoval na náklon, bylo možno přes program RealTerm posílat příkazy, číst a zapisovat do registru akcelerometru. Cílem této fáze bylo porozumění způsobu komunikace akcelerometru s převodníkem.

Další fáze byla vytvoření desky plošných spojů s vyhodnocovacím prvkem a bezdrátové komunikační části, zbavil jsem se tak drátů a komunikačního modulu MAX232. Na desce je místo modulu MAX232 ZigBee modem PAN4551. Drátovou komunikaci jsem nahradil bezdrátovou technologií ZigBee. Vytvořil jsem vizualizační program pro signalizaci a uložení informace o převrnutí do počítače.

V hardwarové části jsou značné možnosti dalšího zlepšení, například vhodné jsou tenzometry k určení zda vozíčkář sedí ve vozíku. Dále je zajímavý monitoring životních funkcí invalidy přímo na vozíku, při zdravotním závážení by se vyslala zpráva do řídicí místnosti. Díky dostupnosti GPS modulů by se vozíčkář dal snadně vystopovat.

Výsledkem práce je zařízení kompaktních rozměrů aby se dalo snadně přidělat k invalidnímu vozíku. Po převrnutí zařízení vyhodnotí alarmový stav a vyšle informaci do přijímací stanice. Alarm se zobrazí v přehledném vizualizačním programu na obrazovce počítače. Toto zařízení se může stát součástí systému vzdálené domácí péče.

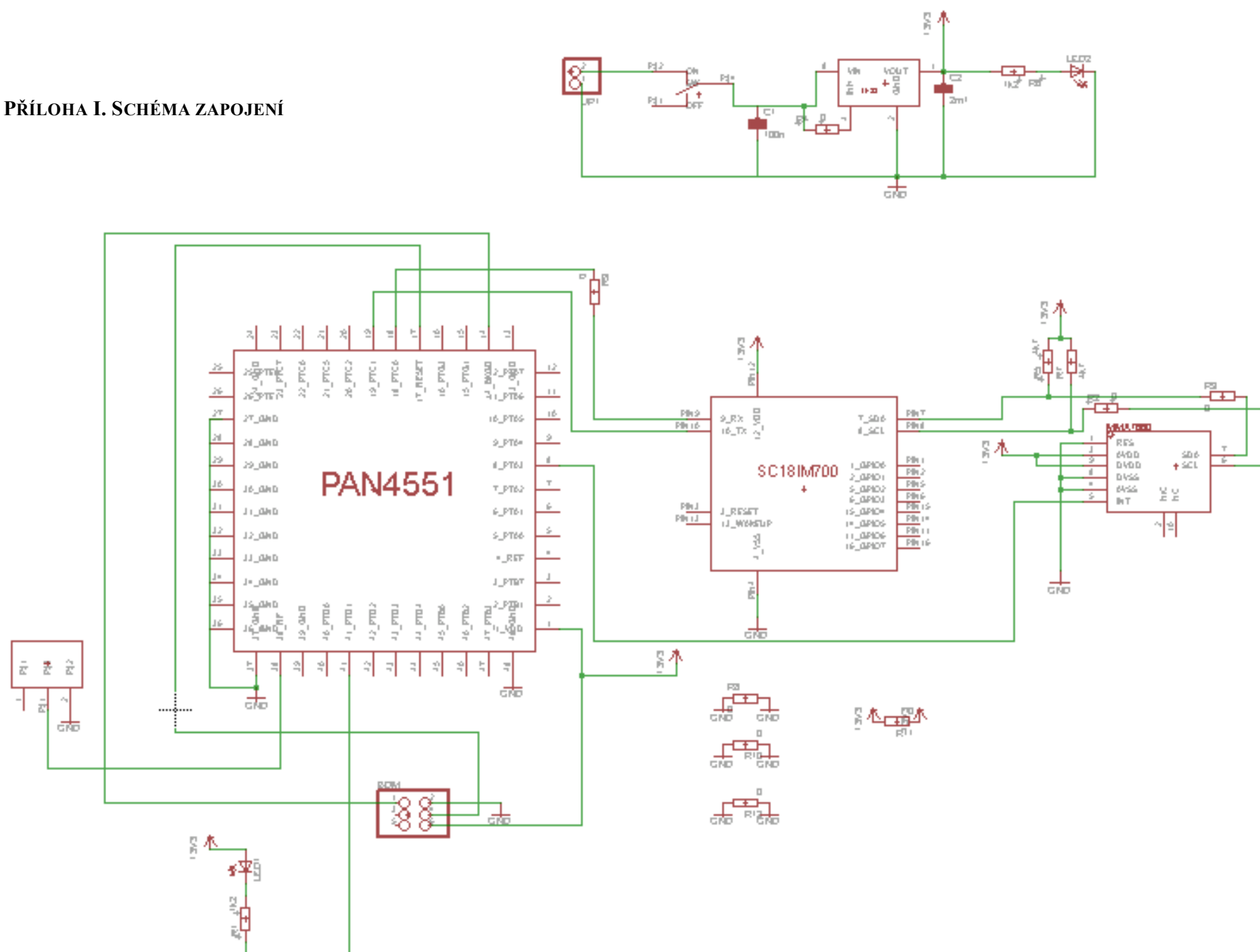
## Seznam použité literatury

- [1] *Akcelerometry* [online]. <<http://wiki.kn.vutbr.cz/robot/index.cgi?akcelerometr> >
- [2] Vojáček, Antonín. *Jak pracují nové 3D MEMS akcelerometry Freescale* [online] <<http://hw.cz/Produkty/Nove-soucastky/ART1875-Jak-pracuji-nove-3D-MEMS-akcelerometry-Freescale-.html> >
- [3] *Gyroskop* [online] <<http://jonatan.spse.pilsedu.cz/~mazanec/gyroskopy.htm>>
- [4] *I<sup>2</sup>C* [online]. <[http://cs.wikipedia.org/wiki/I<sup>2</sup>C](http://cs.wikipedia.org/wiki/I%C2%B2C)>
- [5] Bradáč, Zdeněk. *Bezdrátový komunikační standard ZigBee* [online] <<http://www.automatizace.cz/article.php?a=638>>
- [6] Jedlička, Tomáš. *RealTerm - Sériový terminál* [online] <<http://hw.cz/Teorie-a-praxe/Software/ART881-RealTerm---Seriovy-terminal.html>>
- [7] *Datasheet for MMA7660 3-Axis Orientation/Motion Detection Sensor*  
Freescale Semiconductor, Inc. 2009
- [8] *Datasheet for LE33 VERY LOW DROP VOLTAGE REGULATORS WITH INHIBIT*  
STMicroelectronics, 2005
- [9] *Datasheet for MC9S08GT60A HCS08 Microcontrollers*  
Freescale Semiconductor, Inc. 2008
- [10] *Datasheet for EvalKit Evaluation Board for PAN4551*  
Panasonic, 2005
- [11] *SC18IM700Product Datasheet*  
Philips, 2007
- [12] *Inklinometry a akcelerometry* [online]  
<http://measure.feld.cvut.cz/groups/edu/sz2/pdf/inklinometr.pdf>
- [13] *Rtuťové spínače* [online] <[http://danyk.wz.cz/hg\\_sp.html](http://danyk.wz.cz/hg_sp.html)>
- [14] Vojáček, Antonín. *Principy akcelerometrů 1.díl* [online]. <<http://automatizace.hw.cz/clanek/2007011401>>
- [15] Panasonic Corporation of North America. *ZigBee – Modem PAN4551* [online]. <[www.panasonic.com/industrial/components/pdf/PAN4551.pdf](http://www.panasonic.com/industrial/components/pdf/PAN4551.pdf)>

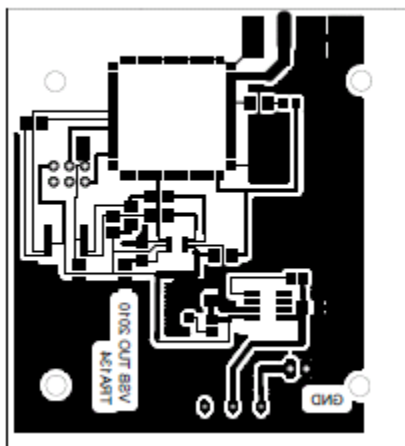
## Seznam příloh

PŘÍLOHA I.	SCHÉMA ZAPOJENÍ .....	- 1 -
PŘÍLOHA II.	DESKA PLOŠNÉHO SPOJE.....	- 1 -
PŘÍLOHA III.	ZDROJOVÝ KÓD PŘIJÍMACÍ STANICE ZIGBEE.....	- 2 -
PŘÍLOHA IV.	ZDROJOVÝ KÓD SENZORU ZIGBEE .....	- 2 -
PŘÍLOHA V.	VÝVOJOVÝ DIAGRAM SOFTWARE NA PŘIJÍMACÍ STANICI.....	- 1 -
PŘÍLOHA VI.	VÝVOJOVÝ DIAGRAM SOFTWARE NA MĚŘÍCÍ DESCE.....	- 1 -
PŘÍLOHA VII.	ZDROJOVÝ KÓD VIZUALIZAČNÍHO SOFTWARE.....	- 2 -

# PŘÍLOHA I. SCHÉMA ZAPOJENÍ



## PŘÍLOHA II. DESKA PLOŠNÉHO SPOJE



Deska plošného spoje (zrcadlově otočeno)

### PŘÍLOHA III. ZDROJOVÝ KÓD PŘIJÍMACÍ STANICE ZIGBEE

```

/*****
*****
*
application template.c

Author: MOTOROLA

Application Note: base RS232
*****
*****
*/

#include <hidef.h> /* for EnableInterrupts
macro */
#include "device_header.h"
#include "pub_def.h"
#include "simple_mac.h"
#include "mc13192_hw_config.h"
#include "application.h"
#include "SCI.h"
#include "mcu_hw_config.h"
#include "drivers.h"

/* Global Variables */
UINT8 gu8RTxMode; /* needed for s-mac,
application can read this variable */
INT8 gi8AppStatus = 0;

void main(void){

/* Variable for wireless communication */
tRxPacket sRxPacket;
tTxPacket sTxPacket;
UINT8
au8RxDataBuffer[10],au8TxDataBuffer[10],accep
t[1],numFlat[4];
/* Other variable */
UINT8
j=0,i=0,end=1,temp=0xFF,numSensor=0,data=0,t1
=0;
UINT16 t2=0;
/* Variable for RS232 communication*/
```

```

extern UINT8 gu8SCIData[5];

MCUInit();
MC13192Init();
SCIInit();

/*Output*/
LED1DIR = 1;
LED2DIR = 1;
LED3DIR = 1;
LED4DIR = 1;

/* Init LED's */
LED1 = 1; /* Default is off */
LED2 = 1;
LED3 = 1;
LED4 = 1;

/*Initialize the packet.*/

sRxPacket.u8DataLength = 0;
sRxPacket.pu8Data
&au8RxDataBuffer[0];
sRxPacket.u8MaxDataLength = 100;
sRxPacket.u8Status = 0;

MLMESetMC13192ClockRate(0); /* Set
initial Clk speed */
UseExternalClock(); /* switch clock
sources */

MLMEMC13192PAOutputAdjust(MAX_POWER
); /* Set MAX power setting */
MLMESetChannelRequest(0);

////////////////////////////////////
////////////////////////////////////
//////////////////////////////////// BASE PROGRAM
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

MyInit();
```

```

// Heart of this program

for(;;){

    MLMERXEnableRequest(&sRxPacket, 0);
    _asm stop;

    if( ((au8RxDataBuffer[2]) >=
0x0001) && ((au8RxDataBuffer[2]) <= 0xFFFE)
){

        LED3 = 0;
        delay(0x5FFF);
        delay(0x5FFF);

        LED3 = 1;
    }

} }

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////      End    of    main
////////////////////////////////////
////////////////////////////////////

void      MCPSDataIndication(tRxPacket
*sRxPacket){

    if (sRxPacket->u8Status == SUCCESS){

        if ( (sRxPacket->pu8Data[0] ==
0x0002) && (sRxPacket->pu8Data[4] ==
0x0003 ) ){

```

```

SCITransmitArray(&sRxPacket-
>pu8Data[0],5);
    }

    }

////////////////////////////////////
////////////////////////////////////

void MLMEMC13192ResetIndication(){
    gi8AppStatus = RESET_STATE;
}

////////////////////////////////////
////////////////////////////////////

void delay(INT16 temp){
    INT16 i;
    for (i=0; i<temp; i++);
}

////////////////////////////////////
////////////////////////////////////

void MyInit(void){

    int i=0;

    /* Initialisation of STOP mode */

    SRTISC=SRTISC&~0x07; /* Disable wake
up delayr. */
    SPMSC2=SPMSC2&~0x03; /* Enable deep
sleep mode stop3. */
    SOPT_STOPE = 1;

    /* Visualisation by LED of the good
iniatialisatoin */
    for(i=0;i<5;i++){

        LED1 = 0;
        LED2 = 0;
        LED3 = 0;

```



```
LED4 = 0;                                delay(0x5FFF);
                                          }
delay(0x5FFF);

EnableInterrupts; /* Turn on system
interrupts */
}

//
```

## PŘÍLOHA IV. ZDROJOVÝ KÓD SENZORU ZIGBEE

```

/*****
*****
*
application template.c

Author: MOTOROLA

Application Note: capteur
*****
*****
*/

```

```
#include <hidef.h> /* for EnableInterrupts
macro */
#include "device_header.h"
#include "pub_def.h"
#include "simple_mac.h"
#include "mc13192_hw_config.h"
#include "application.h"
#include "SCI.h"
#include "mcu_hw_config.h"
#include "drivers.h"
#include "derivative.h" /* include peripheral
declarations */
```

```
/* Global Variables */
UINT8 gu8RTxMode; /* needed for s-mac,
application can read this variable */
INT8 gi8AppStatus = 0;
```

```
void zpozdeni(void){
unsigned int i = 0;
for(i = 0; i< 350; i++){
asm{
nop
nop
nop
nop
nop
}
```

```

    nop
}
}

}

void sendUart2(unsigned char znak){
    SCI2D = znak;
    while(SCI2S1_TDRE==0){
        ;
    }
    zpozdeni();
}

```

```
void main(void) {
    int Preklopeni = 0;
    int Ustaleni = 1000;
    int Detekce = 1;
    INT16 i;
    unsigned int znak;
```

```
tRxPacket sRxPacket;  
tTxPacket sTxPacket;  
UINT8 au8RxDataBuffer[20];  
UINT8 au8TxDataBuffer[20];  
UINT8 flag=0;;
```

```
MCUInit();
MC13192Init();
// SCIIInit();
```

```

/*Initialize the packet.*/
sTxPacket.u8DataLength = 0;
sTxPacket.pu8Data
=&au8TxDataBuffer[0];
sRxPacket.u8DataLength = 0;
sRxPacket.pu8Data
=&au8RxDataBuffer[0];
sRxPacket.u8MaxDataLength = 100;
sRxPacket.u8Status = 0;

```

```

//MLMSetMC13192ClockRate(0); /* Set
initial Clk speed */
//UseExternalClock(); /* switch clock
sources */
MLMEMC13192PAOutputAdjust(MAX_P
OWER); /*Set MAX power setting*/
MLMSetChannelRequest(0);

```

```

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////  PROGRAM OF THE
SENSOR //////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

```

```

au8TxDataBuffer[0] = 0x02;
au8TxDataBuffer[1] = 0x80;
au8TxDataBuffer[2] = 0xff;
au8TxDataBuffer[3] = 0xff;
au8TxDataBuffer[4] = 0x03;

```

```

//EnableInterrupts; /* enable interrupts */
/* include your code here */
SOPT_COPE = 0; //Vypnuti watch dogu

```

```

PTADD_PTADD3 = 0; //Akcelerometr
PTAPE_PTAPE3 = 1;

```

```

PTADD_PTADD4 = 0;
PTAPE_PTAPE4 = 1; //Tlacitko 1

```

```

PTDDD_PTDDD4 = 1;
PTDDD_PTDDD0 = 1;
PTDDD_PTDDD1 = 1;
PTDDD_PTDDD3 = 1; //Dioda
PTDD_PTDD4 = 1;

```

```

PTDD_PTDD0 = 1;
PTDD_PTDD1 = 1;
PTDD_PTDD3 = 1; //Vypnuti led

```

```

for(i=0;i<5;i++){

```

```

    PTAD_PTAD3 = 0;
    delay(0x5FFF);
    delay(0x5FFF);

```

```

    PTAD_PTAD3 = 1;
        delay(0x5FFF);
    delay(0x5FFF);
}

```

```

SCI1BD = 26;
SCI1C1 = 0b00000000;
SCI1C2 = 0b00001100; //Bez preruseni
SCI1C3 = 0b00000000;

```

```

SCI2BD = 26;
SCI2C1 = 0b00000000;
SCI2C2 = 0b00001100; //Bez preruseni
SCI2C3 = 0b00000000;

```

```

sendUart2(0x53);
sendUart2(0x98);
sendUart2(0x02);
sendUart2(0x07);
sendUart2(0x00);
sendUart2(0x50);

```

```

sendUart2(0x53);
sendUart2(0x98);
sendUart2(0x02);
sendUart2(0x07);
sendUart2(0x01);
sendUart2(0x50);

```

```

PTDD_PTDD3 = 0;
delay(0x5FFF);

```

```

for(;;) {

```

```

    sendUart2(0x53);
    sendUart2(0x98);

```



```

delay(0x5FFF);

/* Visualisation by LEDS of the good
initialisation */
    INT16 i;
    for(i=0;i<5;i++){

        PTAD_PTAD0 = 0;
        delay(0x5FFF);
        delay(0x5FFF);

        PTAD_PTAD0 = 1;
        delay(0x5FFF);

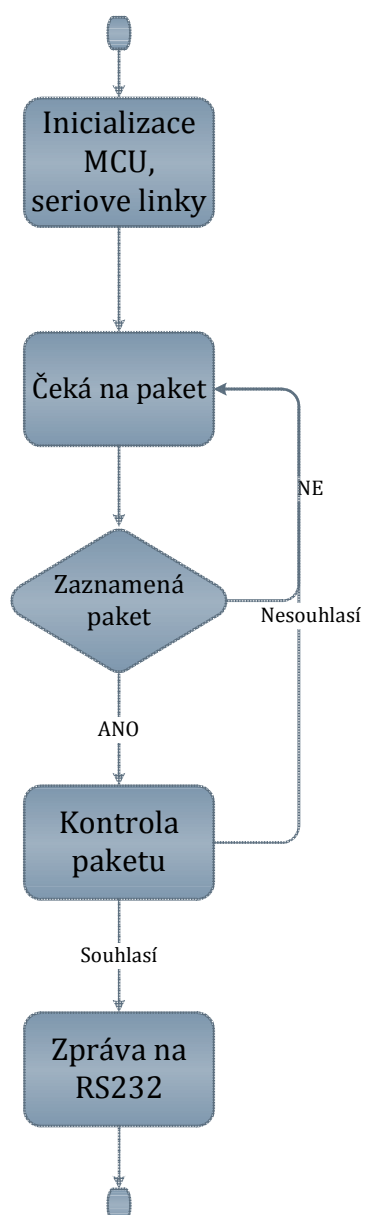
        EnableInterrupts; /* Turn on system
interrupts */

    }

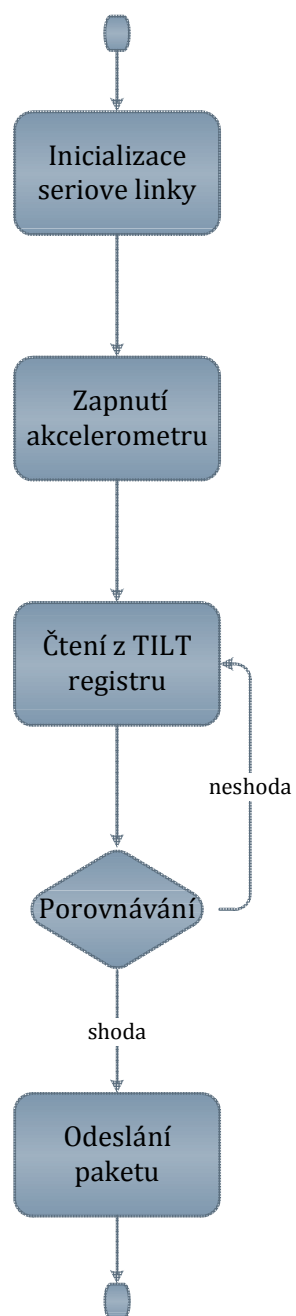
    //////////////////////////////////////
    //////////////////////////////////////

```

## PŘÍLOHA V. VÝVOJOVÝ DIAGRAM SOFTWARE NA PŘIJÍMACÍ STANICI



## PŘÍLOHA VI. VÝVOJOVÝ DIAGRAM SOFTWARE NA MĚŘÍCÍ DESCE



## PŘÍLOHA VII. ZDROJOVÝ KÓD VIZUALIZAČNÍHO SOFTWARE

```
function varargout = BCmatlab(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @BCmatlab_OpeningFcn, ...
                  'gui_OutputFcn',  @BCmatlab_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


function BCmatlab_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);


function varargout = BCmatlab_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;


function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function VolbaCOM_Callback(hObject, eventdata, handles)


function VolbaCOM_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function pushbutton1_Callback(hObject, eventdata, handles)
try
    switch (get(handles.VolbaCOM,'Value'))
        case 1
            OznaceniPortu='COM1';
```



```

        case 2
            OznaceniPortu='COM2';
        case 3
            OznaceniPortu='COM3';
        case 4
            OznaceniPortu='COM4';
        case 5
            OznaceniPortu='COM5';
        case 6
            OznaceniPortu='COM6';
        case 7
            OznaceniPortu='COM7';
        case 8
            OznaceniPortu='COM8';
        case 9
            OznaceniPortu='COM9';
        case 10
            OznaceniPortu='COM10';
    end
    handles.s = serial(OznaceniPortu);
    global ZastaveniCyklu;
    cas=0;
    ZastaveniCyklu=0;

    set(handles.s, 'BaudRate', 9600, 'Parity', 'none', 'Terminator', '', 'Timeout', 0.05);
    fopen(handles.s);

    while(ZastaveniCyklu~=1)
        Vystup=fread(handles.s)
        if (~isempty(Vystup))
            set(handles.s1, 'Value', 0);
            set(handles.s2, 'Value', 1);
            cas=int2str(clock)
        else
            set(handles.s1, 'Value', 1);
            set(handles.s2, 'Value', 0);
        end
        pause(0.5);
    end
    fclose(handles.s);
    file=fopen('cas.txt', 'w+')
    fprintf(file, 'Cas prevrhnuti:\n\n')
    fprintf(file, 'rok:')
    fprintf(file, '   mes: ')
    fprintf(file, '   dat:')
    fprintf(file, '   Hod.')
    fprintf(file, '   min. ')
    fprintf(file, 'sec. \n\n')
    fprintf(file, '%s', cas)
    fclose(file);
    set(handles.s1, 'Value', 0);
    set(handles.s2, 'Value', 0);
    ZastaveniCyklu=0;
    catch
    disp('chyba');
    end
    guidata(hObject, handles);

function pushbutton2_Callback(hObject, eventdata, handles)
global ZastaveniCyklu;
ZastaveniCyklu=1;
guidata(hObject, handles);

```